NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ULTRA-WIDEBAND RADAR TRANSIENT DETECTION USING TIME-FREQUENCY AND WAVELET TRANSFORMS

by

William A. Brooks, Jr.

December 1992

Thesis Advisor: Monique P. Fargues

Approved for public release; distribution is unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Electrical and Computer Engineering Dept.
Naval Postgraduate School

6b. OFFICE SYMBOL *(if applicable)*
EC/FA

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(if applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO

11. TITLE *(Include Security Classification)*
ULTRA- WIDEBAND RADAR TRANSIENT DETECTION USING TIME-FREQUENCY AND WAVELET TRANSFORMS (U)

12. PERSONAL AUTHOR(S)
William A. Brooks, Jr.

13a. TYPE OF REPORT
Master's Thesis

13b. TIME COVERED
FROM ______ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
December 1992

15. PAGE COUNT
128

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Detection of weak ultra-wideband (UWB) radar signals embedded in non-stationary interference presents a difficult challenge. Classical radar signal processing techniques such as the Fourier transform have been employed with some success. However, time-frequency distributions or wavelet transforms in non-stationary noise appears to present a more promising approach to the detection of transient phenomena. In this thesis, analysis of synthetic signals and UWB radar data is performed using time-frequency techniques, such as the short time Fourier transform (STFT), the Instantaneous Power Spectrum and the Wigner-Ville distribution, and time-scale methods, such as the a trous discrete wavelet transform (DWT) algorithm and Mallat's DWT algorithm. The performance of these methods is compared and the characteristics, advantages and drawbacks of each technique are discussed.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Prof. Monique P. Fargues

22b. TELEPHONE *(Include Area Code)*
(408) 646-2859

22c. OFFICE SYMBOL
EC/FA

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited.

Ultra-Wideband Radar Transient Signal Detection Using Time-Frequency and Wavelet Transforms

by

William Allen Brcoks, Jr.
Lieutenant, United States Navy

BSCHE, University of Missouri at Rolla, 1980
MSCHE, University of Missouri at Rolla, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1992

Author: ______________________________
William A. Brooks, Jr.

Approved by: ______________________________
Monique P. Fargues, Thesis Advisor

G. S. Gill
____
G. S. Gill, Co-Advisor

M. A. Morgan
____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

Abstract

Detection of weak ultra-wideband (UWB) radar signals embedded in non-stationary interference presents a difficult challenge. Classical radar signal processing techniques such as the Fourier transform have been employed with some success. However, time-frequency distributions or wavelet transforms in non-stationary noise appears to present a more promising approach to the detection of transient phenomena. In this thesis, analysis of synthetic signals and UWB radar data is performed using time-frequency techniques, such as the short time Fourier transform (STFT), the Instantaneous Power Spectrum and the Wigner-Ville distribution [1], and time-scale methods, such as the *à trous* discrete wavelet transform (DWT) algorithm [2] and Mallat's DWT algorithm [3]. The performance of these methods is compared and the characteristics, advantages and drawbacks of each technique are discussed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. PROBLEM STATEMENT

The Research Development Test and Evaluation Division of the Naval Command Control and Ocean Surveillance Center (NCCOSC RDT&E Division) in San Diego, California has designed and built an ultra-wideband (UWB) radar system to investigate the utility of this technology. The initial research concentrated on the detection of small boats located on a radar sea range. The target returns from small boats consist of short duration transients embedded in sea clutter, multipath and non-stationary background noise. The goal of this thesis is to process the UWB radar returns using time-frequency distribution and wavelet transform spectral analysis techniques for the purpose of target detection.

Classical time-frequency spectral analysis methods can be used for non-stationary signal analysis and are derived from the Fourier transform. To determine the time dependence of the frequency content of a signal, these techniques segment the data through the use of a finite analysis window *g(t)* over which a signal is approximately stationary. The Fourier transform of the windowed data is used to compute the spectrum of the signal as a function of time and, sliding the window along the entire data record results in a time-frequency surface. The use of windows introduces an inherent tradeoff between time and frequency resolution. This tradeoff is a function of the window length. Long windows increase the frequency resolution at the expense of the time resolution and, vice-versa. Thus, these techniques can prove inadequate for analyzing highly non-stationary behavior such as transients. The time-frequency methods discussed in this thesis are the Short Time Fourier Transform (STFT), the Wigner-Ville Distribution (WD) and the Instantaneous Power Spectrum (IPS).

Wavelet transforms can serve as an alternative to conventional time-frequency techniques and may be used in problems where joint resolution in time and frequency are required. Wavelet transforms are similar to windowed Fourier transform but use a stretched or compressed version of the analysis window $g(t/a)$, where a is referred to as the scale factor and is always greater than one. This approach leads to a representation called a time-scale distribution, where the scale varies inversely with frequency. As the scale factor a increases, the analysis window$g(t/a)$ becomes dilated, and the frequency resolution increases. When the scaling factor a decreases, the analysis window is contracted and therefore, the time resolution increases. The scaling properties of wavelet transforms are advantageous in signal processing applications because the transform provides good frequency resolution for signals that are slowly varying in time and provides good time resolution for high frequency signals that are generally highly localized in time.

Time-frequency methods perform their analysis with a constant absolute bandwidth (because the same window is used at all frequencies), while wavelets perform their analysis with a fixed relative bandwidth. This is a primary advantage of time-scale distributions, because these methods allow sharp time resolution at high frequencies (low scales) and sharp frequency resolution at low frequencies (high scales). Thus, this method shows promise for estimating the spectra of UWB radar targets that primarily consist of transient phenomena.

B. OBJECTIVE

The goal of this thesis is to examine time-frequency and time-scale techniques that may be used to detect transient signals originating from small UWB radar targets embedded in non-stationary background noise. Chapter II discusses UWB radar system and the radar signal processing techniques used by the personnel at NCCOSC RDT&E Division. Chapter III examines time-frequency methods such as the Short Time Fourier Transform, the Wigner-Ville distribution and the Instantaneous Power Spectrum. Chapter IV introduces the "Discrete" Continuous Wavelet Transform (DCWT), the a

trous Discrete Wavelet Transform (DWT) algorithm and Mallat's DWT algorithm. The performance of each method on five synthetic test signals and two UWB radar data records is compared in Chapter V. Recommendations and conclusions are presented in Chapter VI. Finally, the MATLAB computer code for each of the time-frequency and time scale methods is presented in the Appendix.

II. THE ULTRA-WIDEBAND (UWB) RADAR PROGRAM AT NCCOSC RDT&E

A. GENERAL DESCRIPTION OF UWB RADAR

An UWB radar has a bandwidth considerably greater than that associated with conventional radar systems. Impulse UWB technology refers to the free space transmission of a short-duration video pulse with a very high peak power and a frequency spectrum that extends from near direct current to several Gigahertz. Hence, UWB radars are also known as "impulse", "non-sinusoidal" or "large fractional bandwidth radars".

Compared to conventional radars, UWB radar is characterized by very large bandwidths and high range resolutions [4]. Non-wideband radars typically operate with a center frequency in the microwave region, have bandwidths on the order of a few Megahertz and pulse widths on the order of a microsecond. Impulse radars may have a center frequency in the UHF region, have a bandwidth of a few hundred Megahertz and have pulse widths on the order of nanoseconds.

The combination of high range resolution, large bandwidth and the low frequencies in UWB radar systems enables this type of radar to detect targets that may not be detected by non-UWB radars. The most potentially useful applications for UWB radars are for detection of target with low radar cross sections (low observables), earth and foliage penetration and, target identification. One disadvantage of UWB radars is the increased signal processing computational burden associated with the high bandwidth which leads to a proportional increase in system cost. This occurs because the number of resolution cells present in a surveillance volume, probability of false alarm, and signal processing load required for target detection all increase with bandwidth. Therefore, UWB radars may be used only when the increased percentage bandwidth presents a distinct advantage over conventional non-UWB radars [4].

Finally, although UWB radars operate with high peak powers, the system delivers a relatively low power per Hertz in comparison to narrow band sources, therefore, background radio frequency interference (RFI) becomes a significant factor to be overcome when processing the radar returns.

UWB radars have bandwidths considerably greater than conventional radars. UWB radar are defined as having a fractional bandwidth ($BW_{fractional}$) greater than 0.25 [4], where the fractional bandwidth is given by:

$$BW_{fractional} = 2(f_h - f_l)/(f_h + f_l) \tag{1}$$

The frequency f_h is the upper bound frequency and the frequency f_l is the lower bound frequency, and 99% of the energy within the signal resides in the frequency band between f_h and f_l. The radar system described in this thesis is an impulse UWB radar with a fractional bandwidth of 1.33.

B. THE NCCOSC RDT&E DIVISION ULTRA-WIDEBAND RADAR PROGRAM

The Research Development Test and Evaluation Division of the Naval Command Control and Ocean Surveillance Center (NCCOSC RDT&E Division) in San Diego, California has designed and built an impulse UWB radar system to explore the applicability and potential of this technology for Naval requirements. The NCCOSC RDT&E Division UWB radar system described in Pollack [5] was built in support of these objectives, and was used to collect data on targets in the presence of sea clutter, multipath and background RFI. This UWB radar operates between 200 and 1000 MHz and the transmitted waveform consists of a single monocycle. Table 1 is a listing of the specifications of the NCCOSC RDT&E Division facility.

The UWB radar is a bistatic system consisting of two thirty foot parabolic antennas overlooking the Pacific Ocean at Point Loma in San Diego, California. Figure 1 is a schematic of the radar transmitter and receiver. Data was collected on severai different

TABLE 1: NCCOSC UWB RADAR SPECIFICATIONS

Bandwidth	200-1000 MHz
Center Frequency	600 MHz
PRF	50 Hz
Pulse Width	2 ns
Peak Power	180 MW (pulser)
Average Power	0.5 W (pulser)
Peak Power	24 KW (radiated)
Minimum Range	5 m
Range Resolution	0.19 m
Design Detection Radar Cross Section	0.001 - 1 m^2

targets, however only records containing target data for a small boat with a small triangular trihedral corner reflector, and a small boat without a corner reflector are considered in this thesis. In each case the target was approximately 1.86 Km down range at an elevation of -3.3 degrees and the data was collected at ten pulses per second with no on-line processing.

To detect targets in the presence of background noise, four signal processing approaches were investigated and are described in detail in Pollack [5]. First, consecutive pulses were averaged, second a matched filter was implemented, third a windowed Fast Fourier Transform (FFT) was utilized, and finally undesirable RFI carriers were excised . The figure of merit used in the analysis was the signal-to-RFI ratio (SRR), which has units of decibels and is defined as the difference between the maximum and the minimum peak in the signal divided by the root mean square value of the RFI. Table 2 is a summary of the SRR improvement for each of the four processing

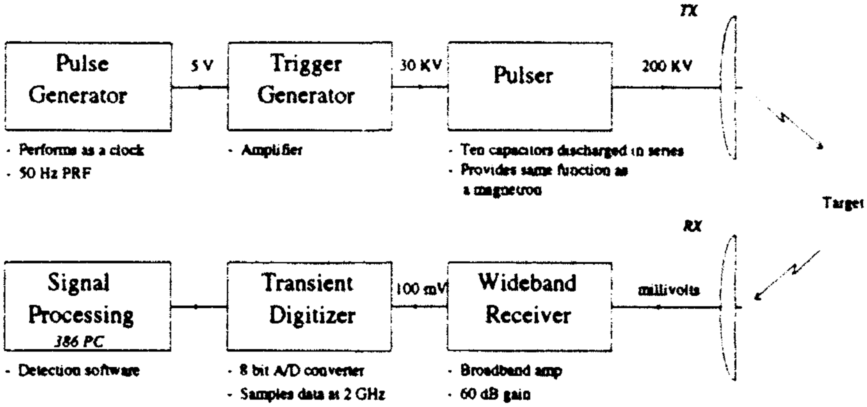


Figure 1. The NCCOSC RDT&E Division UWB radar system

methods for records containing data for the boat with the corner reflector.

TABLE 2: SUMMARY OF SRR IMPROVEMENT	
Matched Filtration	4-5 dB
Excision	13 dB
Averaging Consecutive Pulses	0-25 dB
Sum of Spectral Changes	34 dB

The technique of averaging consecutive pulses improves the SRR by capitalizing on the semi-random nature of the RFI. The method sums the target returns semi-coherently, however, the RFI is summed non-coherently, thereby increasing the SRR. The greatest amount of reduction occurred after 40 pulses were averaged. Further averaging improved the SRR only marginally due to the non-random nature of the RFI at the Point Loma site.

Pulse averaging is a powerful signal processing tool, but suffers from several drawbacks. First, the technique will not average out non-stationary noise. Second, if N averages are required to achieve a satisfactory SRR, then the effec.ive pulse repetition frequency (PRF) is decreased by a factor of N. This effect is a problem because the

UWB system has a maximum PRF of 100 Hz, and a minimum PRF of 50 Hz is required to oversample the sea clutter at the Nyquist rate [5]. If the system is operated at the maximum PRF then at most two pulses may be averaged in order to maintain the Nyquist sampling rate criteria.

The matched fiiter is often used in radar signal processing to maximize the output signal to noise ratio and it is the optimal technique for the detection of known signals in a background of white Gaussian noise [6]. Ideally, this occurs when the magnitude of the matched filter frequency response function $|H(\omega)|$ is equal in magnitude to the spectrum of the reflected signal $|S(\omega)|$, and the phase spectrum of the matched filter is a reversed version of the received signal. The matched filter for the UWB system was obtained by pointing, boresight to boresight, the 30 foot transmitting and 30 foot receiving antennas directly at each other at a range of 70 feet. The radar system transfer function was measured directly by digitizing the output waveform which was used to implement the matched filter. The filtered output data records were computed by convolving the raw data records with the matched filter response.

This method increased the SRR by approximately 5.4 dB. The poor processing gains achieved with this method may be attributed to two reasons. First, the performance of the matched filter is not an optimal filter due to the non-Gaussian nature of the RFI at the Point Loma radar site. Second, the radar system transfer function may have been distorted because the measurements were perforn.ed in the near field region of the receiving and transmitting antennas.

The sum of spectral changes technique is a variation of the short time Fourier transform (STFT) method discussed in depth in Chapter III. In short, this method is a spectral analysis technique that is implemented by sliding a 16 point rectangular window across the 1024 point data record one point at a time. Note, longer windows give better frequency resolution but tend to smoothen non-stationarities and, shorter windows provide better temporal resolution at the expense of frequency resolution. At each window *j*, the spectrum is obtained by taking the fast Fourier transform (FFT) and then plotting the data on a time-frequency surface. To differentiate the UWB signal from the noise in each time interval, the magnitude of the spectrum in window *j* is subtracted from the spectral magnitude in window *j+1*. Finally, in order to compute the amount the spectrum variation from window to window, the spectral changes are summed across all frequency bins. The result is a one dimensional plot of spectral change versus time that provides an indication of how the spectrum of the RFI differs from the spectrum of the RFI and target. This method requires the following assumptions concerning the UWB waveform and return signal immersed in RFI:

1.) The RFI signal is stationary over the duration of the record.
2.) The transmitted waveform is much shorter in duration of the digitized data record.
3.) The return is pure RFI if no target is present.
4.) The received signal is the sum of the RFI and the reflection from the target.
5.) Only a few point targets exist within the window.

This method provided a 34 dB processing gain for the boat and corner reflector data record but was unable to discriminate the target without the corner reflector [5]. The poor processing gain for the record without the corner reflector may have occurred because the first assumption may not be valid. The target could not be differentiated because the RFI is not stationary from window to window, and could not be suppressed by subtracting the spectrum of adjacent windows.

The last signal processing method used was excision of undesirable RFI carriers. This technique takes the FFT of the data record and zeroes out the frequency bins of the carriers containing the maximum spectral magnitudes. After the carriers are excised, the altered spectrum is then transformed back to the time domain. For 16 excisions, this method provided the best results. A maximum processing gain of 12.65 dB was obtained for the data corresponding to the boat with the corner reflector. However, no significant improvement was obtained for the data corresponding to the boat without the corner reflector. The performance of this technique degraded after the excision of 16 carriers due to the undesired removal the target spectrum.

With the exception of pulse averaging, each of the methods discussed above were performed only on the first pulse of 172 pulse data record and may not reflect trends for all pulses. In addition, the techniques perform adequately for the boat with corner reflector but did not adequately suppress the non-stationary background interference noise for the records without the corner reflector.

III. GENERALIZED TIME-FREQUENCY DISTRIBUTIONS

A. TIME-FREQUENCY DISTRIBUTION GENERAL DESCRIPTION

For band limited, wide sense stationary random process *x(t)*, the power spectral density (PSD) of the process is related to the autocorrelation function $R_{xx}(\tau)$ of the process by the Wiener-Khinchine theorem [7]:

$$P_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau. \tag{2}$$

For finite data sets of time interval T the PSD is expressed as:

$$\hat{P}_{xx}(f) = \int_{0}^{T} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau. \tag{3}$$

The PSD has units of power per Hertz and is bandlimited to $\pm 1/2T$ Hz. In addition, the PSD is a strictly real, positive function with the property $R_{xx}(-\tau) = \overline{R_{xx}}(\tau)$, where the bar over the autocorrelation function indicates the conjugate of that term.

Signal energy can also be expressed as a two dimensional joint function of time and frequency $TF(t,f)$. Time-frequency methods provide a time history of the power distribution within a signal and are valuable tools for characterizing signals whose properties change with time. A comprehensive list of the important properties of valid time-frequency distributions is provided in Cohen [1], however, the following three relationships must hold to make the PSD a true energy distribution. First, the time marginal probability distribution represents the energy density spectrum:

$$\int_{t} TF(t,f) dt = |X(f)|^{2}. \tag{4}$$

Secondly, the instantaneous energy is given by:

$$\int_{f} TF(t,f) df = |x(t)|^{2}. \tag{5}$$

Finally, the total energy of the signal is given by:

$$\int_f \int_t TF(t,f)\,dt\,df = E_x. \qquad (6)$$

B. THE FOURIER TRANSFORM (FT)

The basic method for determining the frequency content of a time domain signal *s(t)* is the Fourier transform:

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt, \qquad (7)$$

where the Fourier transform, *S(f)*, contains the frequency information, but lacks temporal information. For finite data sets of length T, the estimated PSD, or periodogram, can be obtained directly from the data by squaring the magnitude of the Fourier transform:

$$\hat{P}_{ss}(f) = \left| \int_0^T s(\tau) e^{-j2\pi f\tau} d\tau \right|^2 . \qquad (8)$$

Thus, the periodogram is a one-dimensional spectral analysis tool that calculates the relative intensity of each frequency component. The methods based on the presumption of local stationarity within the signal and is satisfactory for signals composed of multiple stationary components (e.g., sinusoids) separated by an arbitrary Δf in frequency.

Unfortunately, the basic Fourier transform is of limited use for non-stationary signals because the transform does not track temporal variations within the spectrum. For example, the time at which an abrupt change in signal behavior (e.g., due to a transient) occurs is not apparent from a periodogram because the energy is spread across the entire spectrum. As a result the distribution does not provide information concerning the spectral evolution of a signal in time.

C. THE SHORT TIME FOURIER TRANSFORM (STFT)

The Short Time Fourier Transform (STFT) is a method devised to introduce time dependency into the Fourier analysis of a signal. The STFT is a joint function between time and frequency that maps the original time domain signal into a two-dimensional time-frequency surface. This representation is useful because the method provides information on spectral variations that occur as a function of time within a signal.

The time-frequency surface of the spectrogram is obtained by separating the data into contiguous blocks of equal length (or windows) and computing a spectral estimate from each block. Juxtaposing the spectral estimates obtained from adjacent windows results in an estimate of the time-frequency surface. The squared modulus of the time-frequency surface is called a spectrogram. The spectrogram represents a valid PSD that meets the criteria of equations (4-5).

The use of finite time windows in the STFT allows direct association between temporal and spectral behavior of a signal. If significant changes occur faster than the time interval under scrutiny, then the time window can be shortened to increase the time resolution and ensure local stationarity. Shorter windows in time are better able to track non-stationarities, however, such reductions reduce frequency resolution. Conversely, longer windows in time increase frequency resolution and increase temporal distortions.

The STFT uses a sliding window $g(\tau)$ centered at location t:

$$\hat{P}_{ss}(t,f) = \left|\int_0^T s(\tau)\bar{g}(\tau - t)e^{-j2\pi f\tau}d\tau\right|^2 . \tag{9}$$

The spectral estimate provided by the spectrogram is real-valued and positive and assumes local stationarity. The time-frequency resolution is fixed over the entire distribution. The frequency resolution of the time-frequency surface is defined by:

$$\Delta f^2 = \frac{\int_{-\infty}^{\infty} f^2[G(f)]^2 df}{\int_{-\infty}^{\infty}[G(f)]^2 df}, \tag{10}$$

where $G(f)$ is defined as the Fourier transform of the window. The introduction of a sliding window causes smearing along both time and frequency axis. As a consequence, two signals must be separated by Δf in frequency in order to be resolved. Alternatively, the time resolution is:

$$\Delta t^2 = \frac{\int_{-\infty}^{\infty} t^2 [g(t)]^2 dt}{\int_{-\infty}^{\infty} [g(t)]^2 dt}. \tag{11}$$

Two pulses in time can be discriminated only if they are separated by Δt. Resolution in time and frequency cannot be arbitrarily small as their lower product is bounded by the Heisenberg uncertainty principle [8]:

$$\Delta t \Delta f \geq 1/2, \tag{12}$$

which demonstrates the tradeoff between frequency and time resolution. The degree of smearing depends on the type of window employed and windows, such as Gaussian windows, that meet the lower bound of the Heisenberg criteria are especially desirable because they provide the best simultaneous time-frequency resolution. However, a 41 point Chebyshev window with 10 point step size was employed in this thesis. This window was chosen because it provided very little ripple in the pass band and the pass-band has a very sharp roll-off after the cutoff frequency.

D. THE WIGNER-VILLE DISTRIBUTION (WD)

Stationary methods, such as the periodogram and spectrogram, assume slow temporal variations in the signal and use finite analysis windows that segment the data into lengths that approximate local stationarity. Therefore, the data in each segment must contain enough information to characterize the property of interest, without distorting that property. When the assumption of local stationarity is not valid, then the PSD estimations produced by stationary techniques fail to produce an accurate energy distribution of the signal. For a finite data set of length T, the effects of this problem can be minimized through the substitution of a time-dependent autocorrelation function of the form [9]:

$$R_{ss}(t_1,t_1+\tau)=\frac{1}{T}\int_{t_1}^{t_1+T}s(t)\bar{s}(t+\tau)dt \tag{13}$$

into equation (3). Next the following variables t_1 and t_2 are defined:

$$t_1=t-\frac{\tau}{2} \text{ and } t_2=t+\frac{\tau}{2},$$

which are rearranged as:

$$\tau=t_2-t_1 \text{ and } t=\frac{t_2+t_1}{2}. \tag{14}$$

Substitution of these variables into the equation (13) yields:

$$\begin{aligned} R_{ss}(t_2,t_1) &= R_{ss}(t+\frac{\tau}{2},t-\frac{\tau}{2}) \\ &= E(t+\frac{\tau}{2},t-\frac{\tau}{2}). \end{aligned} \tag{15}$$

The Wigner-Ville distribution is derived when equation (15) is substituted into equation (2) and, an instantaneous autocorrelation value is used in the Wiener-Khinchine theorem:

$$WD(t,f)=\int_{-\infty}^{\infty}s(t+\frac{\tau}{2})\bar{s}(t-\frac{\tau}{2})e^{-j2\pi f\tau}d\tau. \tag{16a}$$

The discrete form of equation (16) is:

$$WD(n,f)=2\sum_{k=-\infty}^{\infty}s(n+k)\bar{s}(n-k)e^{-j4\pi kf}. \tag{16b}$$

Note, that the Wigner-Ville distribution is a quadratic time-frequency distribution. In addition, the distribution may be interpreted as the Fourier transform of the instantaneous, symmetrical Wiener-Khinchine autocorrelation function and, the PSD is equal to $|WD(t,f)|$.

The WD distribution is able to accurately represent temporal fluctuations while maintaining good frequency resolution, however at the endpoints of a finite data segment, the method suffers from degraded time-frequency resolution [10]. The primary disadvantage of this method is the existence of cross terms (interference artifacts between the components in a multicomponent signal) in the time-frequency plane that occur as a result of the bilinear properties of the distribution. For example, if a signal $\tilde{s}(t)$ consists of two components $s_1(t)$ and $s_2(t)$, then:

$$\begin{aligned} WD_{\tilde{s}}(t,f) &= WD_{s_1+s_2}(t,f) \\ &= WD_{s_1}(t,f) + WD_{s_2}(t,f) + 2\mathrm{Re}\left[WD_{s_1s_2}(t,f)\right]. \end{aligned} \tag{17}$$

The WD of $s_1(t)$ and $s_2(t)$ are defined as auto-WD or autoterms [11] of the distribution. $WD_{s_1s_2}(t,f)$ is referred to as the auto-WD of the product $s_1(t) \cdot s_2(t)$, and is defined as the cross terms of the distribution. If $s_1(t)$ occurs at time t_1 and frequency f_1 and $s_2(t)$ occurs at time t_2 and frequency f_2, then the autoterms for each component are centered on the time-frequency surface at (t_1, f_1) and (t_2, f_2) respectively. The cross terms are centered in midtime and midfrequency between (t_1, f_1) and (t_2, f_2). Thus, as the number of components increases, an n component signal always has $\binom{n}{2}$ cross terms. As the number of cross terms increases the time-frequency distribution becomes difficult to interpret and the autoterms become less apparent.

The Wigner-Ville distribution is periodic with π, not 2π. As a result, a real signal must be sampled at twice the Nyquist rate or the analytic version of the real valued signal must be used in the WD algorithm to be prevent aliasing.

E. THE INSTANTANEOUS POWER SPECTRUM (IPS)

The instantaneous power spectrum is obtained by defining an averaged autocorrelation function of the following form [12]:

$$R_{xx}(t_2,t_1)=\frac{1}{2}E\left[x(t)\bar{x}(t+\tau)+\bar{x}(t)x(t-\tau)\right]. \qquad (18)$$

This expression is used as a spectral estimator and can be interpreted as the coherent average of two terms [10]. The first term of the autocorrelation function uses only past information, while the other uses only future information.

Substitution of equation (18) into equation (3) gives the continuous form of the IPS distribution [10]:

$$IPS(t,f)=\frac{1}{2}\int_{-\infty}^{\infty}\left[x(t)\bar{x}(t+\tau)+\bar{x}(t)x(t-\tau)\right]e^{-j2\pi f\tau}d\tau. \qquad (19)$$

The discrete form of equation (19) is:

$$IPS(n,f)=\frac{1}{2}\sum_{k=-\infty}^{\infty}\left[x(n)\bar{x}(n+k)+\bar{x}(n)x(n-k)\right]e^{-j2\pi kf}. \qquad (20)$$

IPS can be interpreted as the instantaneous cross-energy between the signal $x(t)$ and a filtered version the signal at frequency f and the distribution is a valid estimate of the PSD [10]. The IPS time-frequency surface provides an enhanced spectral representation for multicomponent signals, relative to the WD surface, because the cross terms of the IPS distribution are centered on the autoterms. In addition, IPS also features improved spectral resolution at the signal endpoints and, the minimum sampling rate is the Nyquist rate [10].

IV. THE CONTINUOUS AND DISCRETE WAVELET TRANSFORMS

A. INTRODUCTION

Traditional signal processing techniques rely on variations of the Short Time Fourier Transform (STFT). These methods multiply a signal *s(t)* with a compactly supported window *g(t)* centered around an arbitrary point and compute the Fourier coefficients. The coefficients prcvide an indication of the frequency content of a signal in the vicinity of the arbitrary point. The process is repeated with translated versions of the window until the signal is mapped into a time-frequency surface constructed of the Fourier coefficients obtained at each translation. This process uses a single analysis window featuring a constant time-frequency resolution and is well suited for analyzing signals consisting of a few stationary components with spectral descriptions that evolve slowly with time.

Once a type of window has been chosen for the STFT, then the time-frequency resolution across the time-frequency surface is fixed and a tradeoff between time and frequency resolution is created. This tradeoff is referred to as the Heisenberg inequality [13] and means that one can only trade time resolution for frequency resolution. The net effect of this effect is that classical STFT methods are limited in non-stationary applications because abrupt changes in signal behavior cannot be simultaneously analyzed with long duration windows required for good frequency resolution, and short duration windows required for good temporal resolution.

For non-stationary signal analysis, the wavelet transform produces a time-scale representation that is comparable to the time-frequency representation obtained with the STFT but which is better able to track abrupt changes in signal behavior. The wavelet technique uses a single analysis window which is contracted at high frequencies and is dilated at low frequencies [13]. Although the time-bandwidth product, equation (11), remains constant, this method provides good time resolution at high frequencies and good frequency resolution for low frequencies.

Wavelet transforms are used for problems where joint resolution in time and frequency are required. Applications include speech, image and video compression, singularity characterization and noise suppression in non-stationary signal analysis [13]. Wavelets can also act as bases functions for the solutions of partial differential equations and provide fast algorithms for matrix multiplication [13].

The continuous wavelet transform (CWT) is given by:

$$CWT_s(a,n) = \frac{1}{\sqrt{a}} \int s(t)\bar{g}(\tfrac{t-n}{a})dt \tag{21a}$$

where $s(t)$ is the signal and, $\bar{g}(t)$ is the conjugate of the analysis window $g(t)$, or analyzing wavelet, and may be thought of as a high pass filter. The scale factor a denotes a dilation in time, and n a time translation. The factor $1/\sqrt{a}$ normalizes the expression so that the squared magnitude of the CWT coefficients have units of power per Hertz.

If we define $g_a(t) = g(t/a)/\sqrt{a}$ and $g^{+}{}_{a}(t) = g_a(-t)$ then, equation (21a) may be rewritten as a convolution:

$$CWT_s(a,n) = s(t) * \overline{g^{+}{}_{a}}(t). \tag{21b}$$

Thus, the wavelet operation can be seen as a filtering operation of $s(t)$ with a high pass filter of impulse response $\overline{g^{+}{}_{a}}(t)$. Using the properties of the Fourier Transform (FT), the CWT expression can also be given in the frequency domain:

$$\begin{aligned} CWT_s(a,n) &= \sqrt{a}\int S(\omega)\bar{G}(a\omega)e^{jn\omega}d\omega \\ &= \sqrt{a}\; IFT\left[S(\omega)\bar{G}(a\omega)\right]_{t=n}. \end{aligned} \tag{21c}$$

In order to be considered a valid analyzing wavelet, the function $g(t)$ is required to be zero mean, admissible and progressive [14]. The admissibility condition is defined as:

$$\int_f \frac{|G(f)|^2}{|f|} df < \infty \tag{22a}$$

which implies:

$$\int_t g(t)dt = 0 \tag{22b}$$

and is used to ensure that the transformation is a bounded invertible operator. A progressive wavelet is defined as a complex-valued function that satisfies the admissibility condition and whose Fourier transform equals zero for negative frequencies i.e., $G(\omega) = 0$ for $\omega < 0$.

The CWT can be interpreted as a continuous bank of STFTs with a different bandwidth at each frequency. This behavior occurs because the time resolution of the analyzing wavelet is directly related to the scale a and the frequency resolution of the wavelet is inversely related with scale. Low scales correspond to high frequency components and provide good time resolution. High scales correspond to low frequencies and a comparatively poor time resolution.

In short, the primary difference between the STFT and the wavelet transform is that the basis functions of the STFT have a constant time and frequency resolution over the entire time-frequency surface while wavelet transform has a time and frequency resolution that varies as a function of scale. The differences between the time and frequency resolution for the STFT and CWT are illustrated in Figure 2.

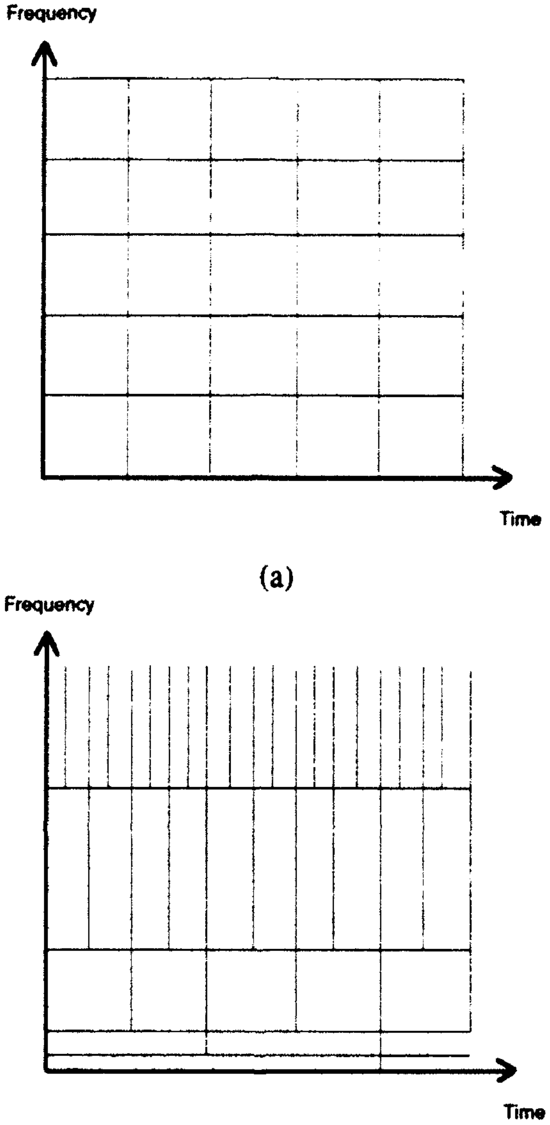
The discrete form of the continuous equation is:

$$DWT(a,n) = \tfrac{1}{\sqrt{a}} \sum_k \bar{g}(\tfrac{k-n}{a})s(k) \tag{23}$$

where the scaling factor a is defined as:

$$a = a_0{}^i \tag{24}$$

and i is an integer number that is termed the octave of the wavelet transform. The factor $a_0{}^i$ indicates that the output at each octave is subsampled by a factor $a_0{}^i$ i.e., the



(a)

(b)

Figure 2. Coverage for the time-frequency plane
(a) for the STFT
(b) for the CWT

frequency resolution at each octave is decreased by a factor of a_0. The choice of a_0 governs the accuracy of the signal reconstruction via the inverse wavelet transform [15]. For most applications $a_0 = 2$ is used because it provides numerically stable reconstruction algorithms and very small reconstruction errors.

Equation (23) is a computationally burdensome form of the wavelet transform because the length of the *DWT* vector doubles for each octave. For example, at the fifth octave the length of the *DWT* vector is 2^5 larger than the original signal [16]. To ease this burden, the decimated version of the wavelet transform was developed [18], [2] and is as follows:

$$DWT(2^i, 2^i n) = \frac{1}{\sqrt{2^i}} \sum_k \bar{g}\left(\frac{k}{2^i} - n\right) s(k). \tag{25}$$

The $2^i n$ term in equation (25) indicates that the length of the output vector at each octave is halved by preserving even points and discarding odd points. This operation keeps the number of *DWT* coefficients constant as the scale increases.

B. DESCRIPTION OF THE DWT ALGORITHMS

Three discrete wavelet transform algorithms are described in the following sections. The "discrete" continuous wavelet transform (DCWT) is an undecimated transform that uses non-orthogonal bases functions i.e., the output is not subsampled by a factor of 2^i , and the analyzing wavelet is admissible, progressive and zero mean. However, the analyzing wavelet does not meet the strict criteria required for orthogonal wavelets outlined in Section E. The *à trous* discrete wavelet transform is a non-orthogonal decimated transform [2], and Mallat's algorithm is an orthogonal, decimated version of the discrete wavelet transform [2].

Non-orthogonal discrete wavelet transform coefficients are not independent and contain redundant information at each octave. Because of their filter properties, non-orthogonal wavelets are desirable because they provide a measure of noise reduction, and have relative bandwidths that mat be controlled by the user. In this thesis, the only non-orthogonal analyzing wavelet considered is the Morlet (modulated Gaussian window) wavelet. The disadvantage of this wavelet is that it is not truly finite in length (not compactly supported) and the original signal may not be reconstructed from the wavelet transform, as only wavelets with finite length filters may be inverted. Orthogonal wavelets are used because they are mathematically elegant, do not contain redundant information wavelets and do lend themselves to signal reconstruction with small reconstruction errors. The major drawback is a lack of flexible filter design that leads to a fixed relative bandwidth of $\pi/2$.

Apart from their filter constraints, the *à trous* algorithm and Mallat's algorithm are identical multiresolution algorithms that may be implemented with filter bank structures [3] that process the signal at different resolutions (r^i) that decrease with increasing octave i. Multiresolution representations are defined as processes that reorganize the signal into a set of details (discrete wavelet transform coefficients) that are computed at each r^i. Each r^i can thought of as a smoothed (low pass filtered) version of the original signal. Given a series of resolutions that decrease with each octave, the wavelet coefficients at each octave are defined as the difference of information between r^i and its approximation at the lower resolution r^{i+1}.

The multiresolution filter bank may be viewed as a two step algorithm of the type shown in Figure 3 (note, s^i denotes the signal at resolution i, the boxes indicate convolution and the down arrow denotes subsampling by a factor of 2). First, the high frequency information is obtained by using the analyzing wavelet g to filter the signal at octave i (s^i). The output of the high pass filtering operation is be referred to as the discrete wavelet transform of the signal at octave i (w^i for the non-orthogonal case and d^i for the orthogonal case). Second, in preparation for the next octave s^i is filtered by the low pass filters, also called scaling functions, denoted by f for the non-orthogonal case or h for the orthogonal case. The output is referred to as the approximated signal at octave $i+1$ (s^{i+1}). This procedure repeats itself as s^{i+1} is filtered by g at the next octave until the detail at each desired octave is computed.

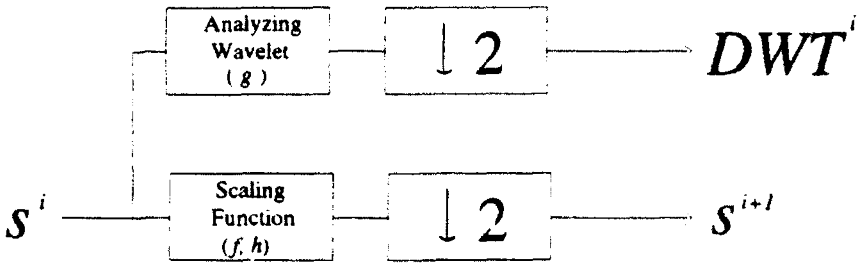


Figure 3. Generic multiresolution filter bank for the DWT

The feature that distinguishes the *à trous* algorithm and Mallat's algorithm is the choice of filters, low pass filters *f* or *h* and, and high pass filters *g*. For the orthogonal wavelets the high pass filter *g* is determined directly from the low pass filter *h*, while for non-orthogonal implementations the high pass filter must only be admissible, progressive and have zero mean and not obtained directly from the low pass filter *f*. In this thesis, only Morlet windows will be used as the high pass filter for the non-orthogonal case.

The *à trous* wavelet transform is a computationally efficient algorithm that computes an exact version of the continuous wavelet transform at discrete points. The method features a relative bandwidth that may be chosen by the user at each octave, but is not invertible (i.e., the original signal cannot be reconstructed from the DWT coefficients) [2]. This occurs because the Morlet wavelet is not a finite filter. Mallat's algorithm has different properties. It computes a discrete approximation of the continuous wavelet transform and is invertible [3] but suffers from a fixed relative bandwidth fixed at $\pi/2$, and therefore has poorer frequency resolution relative to the *à trous* method.

C. THE SCALOGRAM

The spectrogram is defined as the squared modulus of the STFT and provides the energy distribution of a signal with constant resolution on a time-frequency plane. The wavelet spectrogram, or scalogram [13], is defined as the squared modulus of the wavelet transform coefficients $WT(2^i,n)$, and has units of power per frequency unit. The scale-time surface represents a distribution of energy in the time-scale plane.

The scalogram has the same units as the spectrogram but has varying time-frequency resolution. The behavior of a signal on any point on the time axis is localized in the vicinity of the point for small scales. The region of influence of the signal becomes cone-shaped in nature in the time-scale plane as the scale is increased and conversely, the area of localized behavior of a specific frequency on the scalogram shortens as the scale becomes greater.

D. THE NON-ORTHOGONAL DISCRETE WAVELET TRANSFORM

1. The Analyzing Wavelet

The analyzing wavelet used in this analysis is a modulated Gaussian window, or Morlet window [2], of the following form:

$$g(t) = e^{jkt}e^{-\beta^2 t^2/2}. \tag{26}$$

The parameter k is a constant that determines the modulating frequency of the window and β [2] is a constant proportional to the bandwidth of the analyzing wavelet. This type of wavelet was chosen because it meets the lower bound of the Heisenberg criteria [8] and provides optimal resolution in time and frequency [14], [15]. In general, modulated Gaussians are also desirable because their set of linear combinations for pointwise multiplication and convolution is closed and invariant under the Fourier transform. However, the Morlet window is not strictly admissible or progressive because the tail of the Gaussian extends to infinity but, may be forced to approximate these conditions if the window length (L) is on the order of $2\sqrt{2}/\beta$ [2]. For the algorithms in this thesis the relationship $L = 2\sqrt{2}/\beta$ was used for the *à trous* algorithm.

The *à trous* discrete wavelet transform uses the unscaled time domain form of the Morlet window shown in equation (26) in the filter bank implementation of the algorithm. The "discrete" continuous wavelet transform uses a scaled version of the

Fourier transform of the Morlet window described below, however the constraints for β and k outlined below apply to both algorithms.

The Fourier transform of the modulated Gaussian window in the unscaled frequency axis (ω_u) is:

$$\overline{G}(\omega_u) = \tfrac{1}{\beta}\sqrt{2\pi}e^{-\frac{(\omega_u - k)^2}{2\beta^2}}. \tag{27}$$

Frequency scaling is accomplished through he introduction of the scaling parameter a, where $\omega_u = a\omega$:

$$\overline{G}(a\omega) = \tfrac{a}{\beta}\sqrt{2\pi}e^{-\frac{(a\omega - k)^2}{2\beta^2}}. \tag{28}$$

To ensure that $\overline{G}(a\omega)$acts as a highpass filter in the upper half of the spectrum, is admissible and analytic (progressive) and, the spectrum is not aliased, the following restrictions apply to k and β [2]:

$$\pi/2 \le k \le \pi \tag{29}$$

$$\beta \le k/2\pi \tag{30}$$

$$k \le \pi - \sqrt{2}\beta \tag{31}$$

and may be summarized as:

$$\max(2\pi\beta,\ \pi/2) \le k \le \pi - \sqrt{2}\beta. \tag{32}$$

The 3 dB absolute bandwidth of the window is $2\sqrt{2}\beta/a$ and decreases as the number of octaves increases. The relative bandwidth (*RBW*) remains constant for all octaves and is defined as [2]:

$$RBW = \frac{2\sqrt{2}\beta}{k}. \tag{33}$$

The RBW is proportional to β and is constrained by

$$\beta \le RBW \le 2\beta. \tag{34}$$

The frequency resolution may be increased by employing a bank of filters called voices (*M*) that effectively decreases the *RBW*. This process may be thought of as a series of frequency translations of the analyzing wavelet that uses filters of the type *g(t/a)*, with $a = 2^{\frac{j}{M}}$ where *j* varies from 1 to *M*-1. The number of filters, or voices (*M*), in the filter bank is directly proportional to the amount of the upper half of the signal spectrum passed. The number of voices is related to β (i.e., *RBW)* by [2]:

$$M \approx \frac{1}{2\beta} \tag{35}$$

and the windowing function now has the form:

$$g(M) = g\left(\frac{n}{(2^{1/M})^j}\right). \tag{36}$$

The term *j* in the denominator refers to the j^{th} voice out of a total of *M* voices and the bandwidth of the filter at each voice decreases with *j*. As shown in equation (35), an increase in the total number of voices implies a decrease in β or *RBW*, which in turn implies an improvement in frequency resolution. This benefit is offset by the loss of temporal resolution due to the uncertainty principle and an increase of the computational load by a factor of *M* per octave.

2. The "Discrete" Continuous Wavelet Transform (DCWT)

Recall from equation (27) that the CWT of s(t) may be expressed as:

$$\begin{aligned} CWT_s(a,n) &= \sqrt{a}\int S(\omega)\overline{G}(a\omega)e^{jn\omega}d\omega \\ &= \sqrt{a}\, IFT\left[S(\omega)\overline{G}(a\omega)\right]_{t=n}. \end{aligned}$$

where *IFT* indicates the inverse Fourier transform, $a = 2^i$, and $S(\omega)$ is the Fourier transform of the signal *s(t)*. The function $G(a\omega)$ is obtained by replacing the digital frequency in equation (21c) with $\omega = 2\pi f_s / N$ (where f_s is the sampling frequency and *N* is the number of points in the window). The resulting Fourier transform of the sampled discrete Morlet window is given by:

$$\overline{G}[a(\frac{2\pi nf_s}{N})] = \frac{a\sqrt{2\pi}}{\beta}e^{-\frac{(a(\frac{2\pi nf_s}{N})-k)^2}{2\beta^2}}. \tag{37}$$

First, the DCWT algorithm uses of the fast Fourier transform (FFT) to calculate the Fourier transform of the data *s(n)*. Next, the DCWT coefficients at each octave are obtained through the inverse Fourier transform of the product of the transformed window and data record. The code is presented in Appendix A. This method is an undecimated form of the wavelet transform because it preserves all points in the original data sequence. The bandwidth of the window is decreased by a factor of 2^i at each octave and the window length used is 1024 points.

3. The *à* trous Discrete Wavelet Transform

The *à* trous algorithm is a nonorthogonal decimated discrete wavelet transform algorithm proposed by Holscheider et al [17] and first implemented by Dutilleux [16] that is designed to approximate the discrete wavelet series shown in equation (25). As explained earlier, this algorithm is computationally efficient because the number of non-zero DWT coefficients are kept constant as the scale parameter 2^i increases.

This method is used to approximate the nonintegral points of the analyzing wavelet g with an interpolation function f^{+}. The interpolation filter f^{+} is a low pass filter that must satisfy the *à trous* condition:

$$f^{+}(2k)=\delta(k)/\sqrt{2} \tag{38}$$

which means the filter must preserve the even points and discard the odd points of the data sequence. In addition, both f^{+} and g^{+} are both defined as a symmetrical mirror filter with the property that the filter is equal to the conjugate of the time reversed version of itself:

$$f(n)=\overline{f^{+}(-n)}. \tag{39}$$

The unshifted and unconjugated form of the analysis window g in equation (25) may be approximated by the following function [2]:

$$\frac{1}{\sqrt{2}}g^{+}(l/2)\approx\sum_{m}f^{+}(l-2m)g^{+}(m). \tag{40}$$

In short, the interpolating function dilates g^{+} by placing zeros between each pair of coefficients and then the filter f^{+} interpolates the even points to get the odd points.

To derive the à trous algorithm, we set $l = k - 2n$ and write the conjugated form of equation (40) as:

$$\frac{1}{\sqrt{2}}\overline{g^{+}\left(\frac{k-2n}{2}\right)} \approx \sum_{m} \overline{f^{+}(k-2n-2m)g^{+}(m)}. \tag{41}$$

When i is set equal to one, e.g., the first octave, and equation (41) is substituted into equation (25), the result is:

$$DWT(2,2n) \approx \sum_{k}\left[\sum_{m} \overline{f^{+}(k-2n-2m)}\,\overline{g^{+}(m)}\right]s(k). \tag{42}$$

Using the mirror filter properties of f and g, equation (42) can be written as:

$$DWT(2,2n) = \sum_{p} \overline{g^{+}(p-n)} \sum_{k} \overline{f^{+}(k-2p)}s(k) \tag{43}$$

and applying the mirror filter property leads to:

$$DWT(2,2n) = \sum_{p} g(n-p) \sum_{k} f(2p-k)s(k). \tag{44}$$

The term $\sum_{k} f(2p-k)s(k)$ indicates convolution followed by decimation and may rewritten as [2]:

$$\sum_{k} f(2p-k)s(k) = \Lambda(f * s) \tag{45}$$

where Λ indicates subsampling or decimation by a factor of 2^{i} at each octave i. Now, equation (44) may be rewritten in terms of equation (45) as:

$$DWT(2,2n) = [g * (\Lambda(f * s))]_{n}. \tag{46}$$

Equation (46) was derived for i=1, but can be generalized in a two step multiresolution algorithm for $i > 1$ if s is replaced with s^{i}. This leads to the following recursive algorithm:

$$s^{i+1} = \Lambda(f * s^{i}) \tag{47a}$$

$$w^i = g * s^i. \tag{47b}$$

The discrete wavelet transform coefficients w^i (where $w^i \approx DWT(2^i, 2^i n)$) computed by equation (47) are obtained when the filter g high pass filters the upper half of the spectrum of the signal at octave i (s^i). Next, the low frequency information is preserved by the filter f and then decimated to yield the data sequence for the signal at the next octave (s^{i+1}). Note, the analyzing wavelet used in this algorithm is shown in equation (26).

The filter bank implementation of equation (47) is shown in Figure 4a. Note the down arrow indicates the decimation operation and the box indicates the convolution operation. Care must be taken to center the filters f and g to ensure proper alignment of the wavelet coefficients in the scalogram. This concern is illustrated further in Chapter V. Finally, the two choices of *à trous* interpolating filters used in this thesis are [2]:

$$f = [0.5,\ 1,\ 0.5] \tag{48a}$$

and

$$f = \left[-\frac{1}{16},\ 0,\ \frac{9}{16},\ 1,\ \frac{9}{16},\ 0,\ -\frac{1}{16}\right]. \tag{48b}$$

E. THE ORTHOGONAL DISCRETE WAVELET TRANSFORM

1. Mallat's Discrete Wavelet Transform

Mallat's algorithm was originally devised as a computationally efficient method to decompose and reconstruct images [3]. This technique is an orthogonal multiresolution wavelet representation that is used to approximate a signal at a given resolution r_j, and, is also a multiresolution representation that may be implemented in a filter bank structure similar to the *à trous* algorithm. First, let us introduce some new notations. Z and R denote the set of integer and real numbers respectively. The region $L^2(R)$ is defined as a vector space containing the measurable, square-integrable one-dimensional functions $s(x)$ [3]. Next, following Mallat's notation [3] r_j is defined as the resolution, in which the

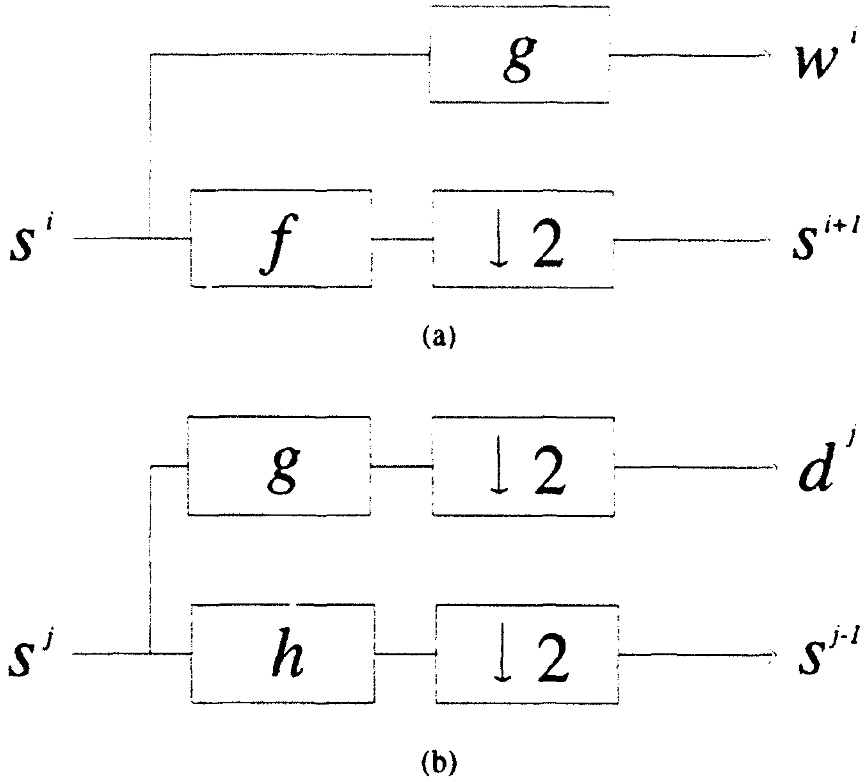


Figure 4. DWT filter algorithms
(a) The *à trous* algorithm
(b) Mallat's algorithm

integer j decreases with increasing scale, not in terms of r_j in which octave j increases with increasing scale (i.e., the resolution is decreased as integer j decreases from zero to $-\infty$ or, as integer j increases from zero to $+\infty$). Finally the signal *s(n)* is defined as *s(x)* in this section to stay consistent with Mallat's notation.

To implement the algorithm in a two step filter bank structure, the signal *s(x)* is first approximated at successive resolutions r_j and r_{j-1} by a low pass filter. Next, a high pass filter is used to extract the detailed information between the approximations of *s(x)* at r_j and r_{j-1}. The low pass and high pass filters are defined as functions $\phi(x)$ and $\Psi(x)$ respectively, and are also referred to as the scaling function and analyzing wavelet. Both the functions $\phi(x)$ and $\Psi(x)$ are members of the orthogonal closed linear subspace

$L^2(R)$. The orthonormal basis used in the decomposition is defined as a family of functions that are built by dilating and translating a unique function $\phi(x)$.

For the special case $r_j = 2^j$, the signal decomposition is achieved by approximating the function $s(x)$ at resolution r_j with the scaling function $\phi(x)$. Thus, the orthonormal basis can be constructed by dilating and shifting the scaling function with a coefficient 2^j. $\left(V_{2^j}\right)_{j\in Z}$ is defined as a family of closed, linear span of subspaces and is the set containing all approximations at resolution 2^j of functions in $L^2(R)$ [3]. The set of vector spaces $\left(V_{2^j}\right)_{j\in Z}$ has the following properties (for $j \in Z$):

$$V_{2^j} \subset L^2(R) \tag{49a}$$

$$\bigcap V_{2^j} = \{0\} \tag{49b}$$

$$\overline{\overline{\bigcup V_{2^j}}} = L^2(R) \tag{49c}$$

where the double bar indicates closure. The space O_{2^j} is defined as the orthogonal complement of the space $\left(V_{2^j}\right)_{j\in Z}$,and both of these spaces are related by:

$$O_{2^j} \oplus V_{2^j} = V_{2^{j+1}}. \tag{49d}$$

A graphical interpretation of these spaces is presented in Figure 5 [18].

If $\left(V_{2^j}\right)_{j\in Z}$ is a multiresolution approximation in $L^2(R)$ then there exists a unique function, or scaling function $\phi(x)$ such that if we define dilated, and dilated and shifted formversions of $\phi_{2^j}(x)$:

$$\phi_{2^j}(x) = 2^j\phi(2^jx) \tag{50}$$

$$\begin{aligned}\phi_{2^j}(x-2^{-j}n) &= 2^{\frac{j}{2}}\phi\left(2^j\left(x-2^{-j}n\right)\right)\\ &= 2^{\frac{j}{2}}\phi(2^jx-n).\end{aligned} \tag{51}$$

Then, the set of scaling functions $\left(2^{-\frac{j}{2}}\phi_{2^j}(x-2^{-j}n)\right)_{n\in Z}$ define an orthonormal basis for V_{2^j} that lies in $L^2(R)$. In addition, the scaling function $\phi(x)$ has the property that the

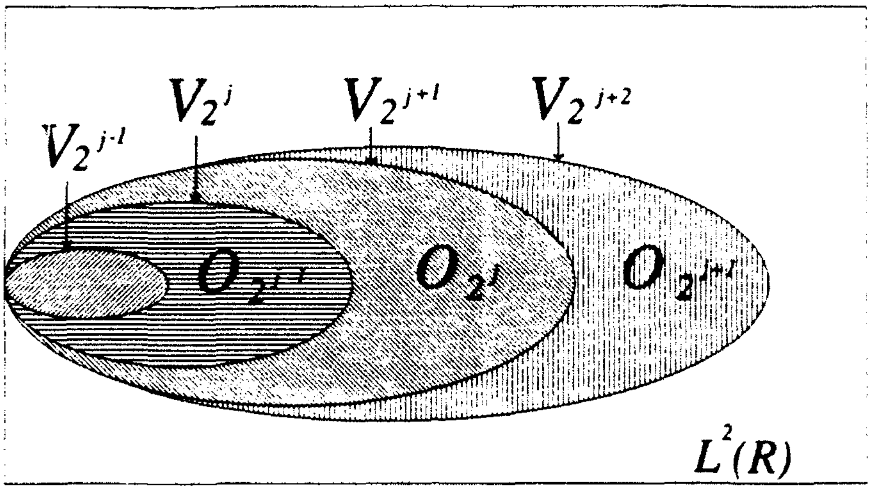


Figure 5. Orthogonal vector subspaces for Mallat's DWT algorithm

version at scale 2^{j} can be approximated by a version of itself at scale 2^{j+1} [3]:

$$\phi_{2^j}(x-2^{-j}n)=2^{-j-1}\sum_k\left\langle\phi_{2^j}(x-2^{-j}n),\phi_{2^{j+1}}(x-2^{-j-1}k\right\rangle\phi_{2^{j+1}}(x-2^{-j-1}k). \tag{52}$$

The inner product *(IP)* in the above expression can be simplified as follows:

$$IP=2^{-j-1}\left\langle\phi_{2^j}(x-2^{j}n),\phi_{2^{j+1}}(x-2^{-j-1}k)\right\rangle$$

$$=2^{-j-1}\int_{-\infty}^{\infty}\phi_{2^j}(u-2^{j}n)\phi_{2^{j+1}}(u-2^{-j-1}k)du$$

$$=2^{j-1}\left[\int_{-\infty}^{\infty}2^{j}\phi(2^{j}u-n)2^{j+1}\phi(2^{j+1}u-k)du\right]$$

$$=2^{j}\int_{-\infty}^{\infty}\phi(2^{j}u-n)\phi(2^{j+1}u-k)du. \tag{53}$$

Using the following substitutions:

$$2^{j+1}u=v$$

$$2^{j+1}du=dv$$

in equation (53) leads to:

$$IP = 2^{j} \int_{-\infty}^{\infty} \phi(2^{-1}v - n)\phi(v-k)2^{-j-1}dv$$

$$= 2^{-1} \int_{-\infty}^{\infty} \phi(2^{-1}v - n)\phi(v-k)dv.$$

$$= 2^{-1} \int_{-\infty}^{\infty} \phi\left(2^{-1}(v-2n)\right)\phi\left(v-k\right)dv. \tag{54}$$

Replacing *w=v-2n* in the above equation leads to:

$$IP = 2^{-1} \int_{-\infty}^{\infty} \phi[2^{-1}(w)]\phi(w+2n-k)dw$$

$$= \int_{-\infty}^{\infty} \phi_{2^{-1}}(w)\phi(w-k+2n)dw$$

Then substituting the expression for IP in equation (52) leads to:

$$IP = \left\langle \phi_{2^{-1}}(w), \phi\left(w-\left(k-2n\right)\right)\right\rangle. \tag{55}$$

$$\phi_{2^{j}}(x-2^{-j}n) = \sum_{k} \left\langle \phi_{2^{-1}}(w), \phi(w-(k-2n))\right\rangle \phi_{2^{j+1}}(x-2^{-j-1}k). \tag{56}$$

Let *h(l)* be defined as the discrete filter with impulse response:

$$h(l) = \left\langle \phi_{2^{-1}}(u), \phi(u-l)\right\rangle \tag{57a}$$

thus for *l=k-2n*:

$$h(k-2n) = \left\langle \phi_{2^{-1}}(w), \phi(w-(k-2n))\right\rangle. \tag{57b}$$

Let h^{+} be defined as the mirror filter with the impulse response $h^{+}(l) = h(-l)$. Replacing h^{+} in equation (57b) we observe:

$$h^{+}(2n-k) = \left\langle \phi_{2^{-1}}(w), \phi(w-(k-2n))\right\rangle. \tag{58}$$

Therefore, equation (52) may be written in terms of $h^{+}(2n-k)$ and the scaling function at 2^{j+1}:

$$\phi_{2^{j}}(x-2^{-j}n) = \sum_{k} h^{+}(2n-k)\phi_{2^{j+1}}(x-2^{-j-1}k). \tag{59}$$

At resolution 2^j, the operator $A_{2^j}s(x)$ is defined as the discrete orthonormal projection of the signal *s(x)* on the orthonormal basis $\left(2^{-\frac{j}{2}}\phi_{2^j}(x-2^{-j}n)\right)_{n\in Z}$ and is characterized by the set of inner products:

$$A_{2^j}s = \left(\left\langle s(x), \phi_{2^j}(x-2^{-j}n)\right\rangle\right)_{n\in Z}. \tag{60}$$

$A_{2^j}s$ is obtained by projecting the function *s(x)* onto the orthonormal basis $\left(2^{-\frac{j}{2}}\phi_{2^j}(x-2^{-j}n)\right)_{n\in Z}$:

$$A_{2^j}s(x) = 2^{-j}\sum_k \left\langle s(x), \phi_{2^j}(x-2^{-j}k)\right\rangle \phi_{2^j}(x-2^{-j}k)$$

Using equations (50) and (58), this leads to:

$$\begin{aligned} A_{2^j}s(x) &= \sum_k h^+(2n-k)\left\langle s(x), \phi_{2^{j+1}}(x-2^{-j-1}k)\right\rangle \\ &= \sum_k h^+(2n-k)A_{2^{j+1}}s(x). \end{aligned} \tag{61}$$

Equation (61) shows that $A_{2^j}s(x)$ may be obtained by convolving $A_{2^{j+1}}s(x)$ with the filter h and keeping every other sample of the output i.e., decimating the output. Thus, $A_{2^j}s(x)$ acts as a linear approximation operator for signal *s(x)* and is used to compute the orthogonal projection of the signal onto the vector space $V_{2^j} \subset L^2(R)$. The vector space V_{2^j} can now be interpreted as the set of all approximations at resolution 2^j of the functions in $L^2(R)$, therefore, $A_{2^j}s(x)$ is the approximation function most similar to *s(x)* in $L^2(R)$. Note that when computing $A_{2^j}s(x)$ at resolution 2^j some information in *s(x)* is lost, but as the resolution is increased ($j \to +\infty$), the approximation converges to the original signal. Thus, equation (61) can be rewritten in a recursion in terms of s, j and the decimation operator Λ:

$$s^{j+1} = \Lambda(h^+ * s^j) \tag{62}$$

where the starting point s^0 is defined as the original sequence s.

The difference of information between two resolutions is defined as the *detail signal* [3] and is the orthogonal projection of s on space the O_{2^j} ,where O_{2^j} is the orthogonal complement of the space V_{2^j}. Therefore the space O_{2^j} contains the detail signal information between $A_{2^j}s$ and $A_{2^{j+1}}s$ in $L^2(R)$.

An orthonormal basis of O_{2^j} is built by dilating and translating a wavelet analyzing function. Following Mallat's development, let $\Psi(x)$ be defined as the wavelet function and let:

$$\Psi_{2^j}(x) = 2^j \Psi(2^j x)$$

denote the dilation of $\Psi(x)$ by 2^j, then:

$$\begin{aligned}\Psi_{2^j}(x-2^{-j}n) &= 2^{\frac{j}{2}}\Psi\left(2^j\left(x-2^{-j}n\right)\right)\\ &= 2^{\frac{j}{2}}\Psi(2^j x-n). \qquad (63)\end{aligned}$$

Now the orthonormal basis $\left(2^{-\frac{j}{2}}\Psi_{2^j}(x-2^{-j}n)\right)_{n\in Z}$ can be expanded as:

$$\Psi_{2^j}(x-2^{-j}n) = 2^{-j-1}\sum_k \left\langle \Psi_{2^j}(u-2^{-j}n), \phi_{2^{j-1}}(u-2^{-j-1}k\right\rangle \phi_{2^{j+1}}(x-2^{-j-1}k). \qquad 64)$$

Similar to the development of *h(l)*, the filter *g(l)* is defined as the discrete filter with the impulse response:

$$g(l) = \left\langle \Psi_{2^{-1}}(u), \phi(u-l)\right\rangle$$

for *l=k-2n* we get:

$$g(k-2n) = \left\langle \Psi_{2^{-1}}(w), \phi(w-(k-2n))\right\rangle. \qquad (65a)$$

Using the mirror filter property $h^{+}(l) = h(-l)$ leads to:

$$g^{+}(2n-k) = \left\langle \Psi_{2^{-1}}(w), \phi(w-(k-2n))\right\rangle. \qquad (65b)$$

Next, substituting equation (65b) into equation (64) yields:

$$\Psi_{2^j}(x-2^{-j}n) = \sum_k g^{+}(2n-k)\phi_{2^{j+1}}(x-2^{-j-1}k). \qquad (66)$$

Let $D_{2^j}s(x)$ be defined as the orthogonal projection of the detail signal on the space O_{2^j} which contains the difference of information between $A_{2^j}s(x)$ and $A_{2^{j+1}}s(x)$. $D_{2^j}s(x)$ is computed by decomposing the signal *s(x)* into the orthonormal basis

$$\left(2^{-\frac{j}{2}}\Psi_{2^j}(x-2^{-j}n)\right)_{n\in Z} \quad :$$

$$\begin{aligned} D_{2^j}s(x) &= \left(\left\langle s(x),\Psi_{2^j}(x-2^j n)\right\rangle\right)_{n\in Z} = \sum_k g^+(2n-k)\left\langle s(x),\phi_{2^{j+1}}(x-2^{-j-1}k)\right\rangle \\ &= \sum_k g^+(2n-k)A_{2^{j+1}}s(x) \\ &= \Lambda(g^+ * s^i). \qquad (67) \end{aligned}$$

Qualitatively, equation (67) shows that the detail signal at each octave can be computed by convolving the signal with the filter *g* and keeping every other sample. This implementation can be visualized by the filter bank structure shown in Figure 4b.

2. The Relationship Between The Analyzing Wavelet and The Scaling Function

Both the *à trous* and Mallat's algorithm are multiresolution algorithms that may be implemented using filter bank structures. These two methods are implemented in the same manner, but differ in the choice of filters *f*, *h* and *g*. In the *à trous* algorithm, the choices of filters *f* and *g* are limited to different sets of criteria that make each a valid filter suitable for this technique. In Mallat's algorithm the filters *h* and *g* are constrained by the orthogonality restrictions explained below and have filter impulse responses directly related to one another. Note that the *à trous* interpolating filters *f* are related to the orthogonal filters *h* by the following relation [2]:

$$\bar{h} * h = f/\sqrt{2}. \qquad (75)$$

The relationship between the Fourier transform of the analyzing wavelet and the Fourier transform of the scaling function is given in Theorem 3 in Mallat [3]:

$$\psi(\omega) = G\left(\frac{\omega}{2}\right)\phi\left(\frac{\omega}{2}\right) \tag{68}$$

with

$$G(\omega) = e^{-j\omega}H(\omega + \pi). \tag{69}$$

In the time domain, the impulse response of the orthogonal filter $G(\omega)$is related to the impulse response of the orthogonal filter $H(\omega)$ by [3]:

$$g(n) = (-1)^{1-n}h(1-n). \tag{70a}$$

and the causal form of equation (70a) is given by [19]:

$$g(n) = (-1)^{n}h(L-1-n). \tag{70b}$$

The filters h and g must have compact support (i.e., zero outside a finite interval) and to ensure orthonormal resolution are constrained by [2]:

$$\sum_{j}[h(2j-n)h(2j-m) + g(2j-n)g(2j-m)] = \delta_{nm} \tag{71a}$$

$$\sum_{j} h(2n-j)g(2m-j) = 0 \tag{71b}$$

$$\sum_{n} g(n) = 0 \tag{71c}$$

$$\sum_{n} h(n) = \sqrt{2}. \tag{71d}$$

Daubechies [15] has discovered an entire family of wavelets that satisies the above conditions. Three examples of this type of filter are the two, four, and twelve point filters shown below:

$$h = \frac{1}{\sqrt{2}}[1,\ 1] \tag{72}$$

$$h = \frac{1}{4\sqrt{2}}\left[1+\sqrt{3},\ 3+\sqrt{3},\ 3-\sqrt{3},\ 1-\sqrt{3}\right] \tag{73}$$

$$\begin{aligned} h = [&0.112,\ 0.494,\ 0.751,\ 0.315,\ -0.226,\ -0.130,\\ &0.98,\ 0.028,\ -0.032,\ 0.005,\ 0.005\ ,\ -0.001]. \end{aligned} \tag{74}$$

V. COMPARISON OF THE ALGORITHMS

A. DESCRIPTION OF THE TEST SIGNALS

The Short Time Fourier Transform (STFT), Wigner-Ville Distribution (WD), Instantaneous Power Spectrum (IPS) and Discrete Wavelet Transform (DWT) algorithms are applied to the following five test signals: an impulse function, a single complex sinusoid, a linear chirp, two crossing linear chirps and two crossing linear chirps in white Gaussian noise (WGN) with a 0 dB signal-to-noise ratio (SNR). The respective equations used for the test signals are:

$$s(n) = \delta(n-256) \tag{76}$$

$$s(n) = e^{j2\pi(0.2)n} \tag{77}$$

$$s(n) = e^{j2\pi(.01)n^2} \tag{78}$$

$$s(n) = e^{j2\pi(.01)n^2} + e^{j2\pi(.01)(1024-n)^2} \tag{79}$$

$$s(n) = e^{j2\pi(.01)n^2} + e^{j2\pi(.01)(1024-n)^2} + \text{WGN}(n). \tag{80}$$

Each record consists of 1024 points. The impulse and single complex sinusoid occurred at bin 256.

B. DESCRIPTION OF THE ULTRA-WIDEBAND RADAR (UWB) SIGNALS

In this thesis, we have investigated UWB radar signal returns from a small boat (with and without a corner reflector) in the presence of sea clutter, multipath and radio frequency interference (RFI). In each case, the small boat was located at approximately 1.86 Km from the radar site at an elevation of -3.3 degrees. The corner reflector was triangular trihedral in shape and was located approximately 10 feet above the surface of the water. For both cases, the experimental data consists of 172 pulse returns, where each pulse return used has a length of 1024 points. The measurements were taken on the same day with approximately the same sea conditions. Only the results for the first

UWB radar pulse in the 172 pulse recorded are included in this chapter, however, the results presented for the first pulse are applicable to all pulses.

C. DESCRIPTION OF THE ALGORITHMS

1. Description of the Time-Frequency Algorithms

For each time-frequency method, a variety of window lengths and step sizes were used, however, the parameters outlined below provide excellent simultaneous time-frequency resolution that result in unambiguous discrimination of the test signals. The STFT algorithm uses a 41 point Chebyshev window with a ten point step size. The WD algorithm used was derived by Parker [20], and was implemented with a 64 point rectangular window with a 32 point step size that provides a 50 % overlap between the sliding windows. The IPS algorithm used was derived by Hagerman [21] and is used with a rectangular window of length 64 and a step sizes of 8 for the synthetic signals, and a 128 Hamming window and step sizes of 4 points for the UWB radar signals.

2. Description of the Time-Scale Algorithms

The following parameters for the time-scale parameters were chosen by trial and error, and provide the best processing gain for the synthetic and UWB data. Note, the time resolution of the DWT is directly related to the scale *a* and the frequency resolution of the wavelet is inversely related with scale. Low scales correspond to high frequency components and provide good time resolution. High scales correspond to low frequencies and a comparatively poor time resolution.

The DCWT method is implemented with a 1024 point Morlet window. The *à trous* DWT algorithm uses the three point interpolating filter (f=[0.5, 1, 0.5]), a Morlet window, and is used with one, five and ten voices. For both methods $k = \pi$ and a β of 0.6 provide the best processing gain for the synthetic data records. To detect the transients in the UWB records, a decrease in bandwidth of the Morlet window to β=0.35 was necessary to detect the target for the radar record return signal corresponding to the boat without corner reflector. Mallat's DWT algorithm is implemented using a Daubechies [15] 12 point orthonormal scaling function *h* and analyzing wavelet *g*. Other combinations of DWT parameters and filter lengths were used, but the listed values provide the best processing gain.

3. Cross Terms

Recall , the primary disadvantage of the Wigner-Ville Distribution described in Chapter III was the existence of cross terms that occur midtime and midfrequency between multicomponent signals. Cross terms also exist for the magnitudes of the coefficients of the STFT, IPS and the wavelet transform time-frequency/scale distributions. Cross terms that occur between closely spaced signals can have significant amplitudes that corrupt the transform spaces of the time-frequency and scale-frequency distributions. Thus, cross terms can provide a serious limitation in the analysis of multicomponent signals.

The STFT, IPS and CWT cross terms will occur at the intersection of two overlapping signals, unlike the WD cross terms which always occur midtime and midfrequency between two WD autocomponents [10],[11]. Thus, for n multicomponent signals, the STFT, IPS and the CWT can have minimum of zero cross terms (for no overlapping signals) or, a maximum of $\binom{n}{2}$ cross terms, unlike a total of $\binom{n}{2}$ cross terms for WD. In addition, the cross terms of the STFT can also have a maximum magnitude equal to twice the product of the magnitude of the spectrograms for each individual signal.

4. Definition of the Processing Gain Ratio (PGR)

Nete that none of the time-frequency or DWT methods described in this thesis actually reduce the sea-clutter, background noise or radio frequency interference (RFI), as did the methods listed in Chapter II. Therefore, the results cannot be described in terms of an increase in signal-to-noise ratio (SNR). The time-frequency and DWT techniques provided in Chapters III-IV serve to enhance the distinguishing characteristics of the desired signal in the presence of undesirable noise. Thus, the results can be described in terms of a processing gain ratio (PGR). The PGR is computed in decibels (dB) and is defined as the voltage ratio of the maximum voltage value ($V_{\max}$) in the time-frequency/time-scale surface divided by the mean voltage value (V_{mean}) of the surface. Therefore, the PGR is described by the following equation:

$$(PGR)_{dB} = 20\log\left(\frac{V_{\max}}{V_{mean}}\right). \tag{81}$$

In addition the PGR is arbitrarily set equal to zero if $V_{\max}$ occurred at time bin less than 100 or at time bin greater than 1000 due to the potential dominance of the noise at the beginning (time bin<100) and the end of the UWB radar data records (time bin>1001). Consequently, a positive PGR was computed if $V_{\max}$occurred between time bins 101-1000. This definition of PGR is not generic and cannot be used for arbitrary targets, but is considered here because the target can only occur between time bins 101-1000 for the experimental data used in thesis.

D. COMPARISON OF THE ALGORITHMS

1. Impulse Function

Figures 6-8 are time-frequency distributions for the impulse function *s(n)*= $\delta(n-256)$ computed by the STFT, WD, and IPS methods. Figures 9-11 are time-scale representations of the impulse function for the DCWT, *à trous* DWT and Mallat's DWT algorithms. The top figure is the contour plot of the magnitude of the two-dimensional surface and the bottom plot is the corresponding three-dimensional mesh plot.

The impulse function is chosen as a test signal because it demonstrates the ability of the various methods to localize a signal in time. Each technique localizes the signal at bin 256. As expected, the time-frequency representations obtained for the STFT, WD and IPS are constant for all frequencies at bin 256, while the time-scale surfaces are cone-shaped in nature. Recall that the scalogram shows a detailed view of the signal in time at high frequencies (small scales) and the global behavior of the signal with increasing scale (low frequencies). The cone-shaped behavior of the WT is better understood by computing the analytic expression of the transform of *s(t)* using equation (21a):

$$
\begin{aligned}
CWT(a,n) &= \frac{1}{\sqrt{a}} \int \delta(t - t_0)\,\bar{g}\left(\frac{t-n}{a}\right) dt, \\
&= \frac{1}{\sqrt{a}}\,\bar{g}\left(\frac{t_0 - n}{a}\right).
\end{aligned}
\tag{82}
$$

Recall that *g(n)* is non-zero over a finite interval because the function is admissible. Thus, as the scaling factor *a* increases, the interval over which CWT*(a,n)* is non-zero increases by a factor of *a*, resulting in the cone-shaped support of the CWT in the vicinity of t_0.

In addition, the time-scale representation of the shifted dirac is used to check the filter alignment for the filters present in the DWT algorithms, as cautioned by Dutilleux [16]. The DWT scaling and analyzing filters are aligned properly because the time-scale representation radiates symmetrically from bin 256 and is not clearly offset in one direction on either side of bin 256.

2. Complex Sinusoid

Figures 12-14 present the magnitudes of the time-frequency distributions and Figures 15-18 are the magnitudes of the time-scale expressions obtained for the complex sinusoid described in equation (77). Note the frequency resolution for the *à trous* DWT algorithm (but not Mallat's DWT algorithm) can be increased by the introduction of multiple voices. The addition of voices serves to decrease the *RBW* of the Morlet analyzing wavelet. In turn, this action increases the length of $g(t)$, and because of the Heisenberg uncertainty principle, the time resolution of the algorithm is decreased.

Again, the representation of an arbitrary complex sinusoid on the time-scale surface is better understood by computing the analytical expression of the wavelet transform of $s(t) = e^{j2\pi\omega_0 t}$. Using equation (21c), the CWT of $s(t)$ is obtained by:

$$\begin{aligned} CWT(a,n) &= \int S(\omega)\overline{G}(a\omega)e^{j\omega n}d\omega \\ &= 2\pi\int \delta(\omega-\omega_0)\overline{G}(a\omega)e^{j\omega n}d\omega \\ &= 2\pi\overline{G}(a\omega_0)e^{j\omega_0 n}. \end{aligned} \tag{83}$$

Therefore, the CWT of a complex sinusoid may be viewed as a modulated version of the Fourier transform of the analyzing wavelet at the modulating frequency ω_0. Thus, the time-scale surface of a complex sinusoid is represented by a frequency band located at the modulating frequency.

The scalograms corresponding to the DCWT, *à trous* algorithm (one voice), and Mallat's algorithm present equivalent frequency resolution, as shown in Figures 15-16 and 18. This frequency resolution may be considered poor when compared to the time-frequency methods. In each of these cases the resolution is not adjustable and it could be difficult to resolve two sinusoids of with similar frequencies. As shown in Figure 17(a) and Figure 17(b), the frequency resolution is adjustable in the *à trous* algorithm through the introduction of five and ten voices.

3. Single Linear Chirp

All time-frequency and time-scale transforms show good time and frequency resolution throughout the range of frequencies. This was expected because only one signal is present and therefore, no cross terms exist for a single linear chirp, as shown in Figures 19-25. Also, the time-frequency methods have a fixed time-frequency resolution over all frequencies, which allows good resolution at low frequencies.

The performance of the DCWT, one voice *à trous* and Mallat's algorithm are comparable. However, their performance is inferior when compared to the STFT, Wigner-Ville and IPS methods for low frequencies (high scales). For five and ten voices, however, the performance of the *à trous* method is comparable to the spectrograms for high frequencies (low scales).

4. Two Crossing Linear Chirps

Time-frequency methods exhibit good time and frequency resolution but suffer from degraded performance due to the presence of undesirable cross-terms near the crossing point of the two chirps, as shown in Figures 26-28. For the STFT and IPS transforms, the cross terms occur on the autoterms of the time-frequency distribution and are additive at the crossing point of the two chirps [10],[11]. The cross terms of the Wigner-Ville distribution appear at the mid-point between the autoterms of the two signals [11] and interfere with the autoterms. In all three cases, the cross terms dominate the plot at the crossing point of the two chirps. The effect causes problem when analyzing multicomponent signals with frequency components that are close together. The cross-terms can be minimized by reducing the size of the window but at the expense of reducing the frequency resolution.

The performance of the DCWT, the *à trous* (one voice), Mallat's algorithm are once again comparable and do not show good frequency resolution. The cross terms occur on the autoterms [11] and also dominate at the midpoint of the two signals. As seen from Figures 31(a)-(b) the freqeuncy resolution can be improved with the addition of voices but the cross-terms are still present at the crossing point of the two chirps.

5. Two Crossing Linear Chirps in White Gaussian Noise (WGN)

Figures 33-39 show time-frequency and time-scale distributions for each of the methods for two crossing linear chirps embedded in white Gaussian noise. The signal-to-noise ratio (SNR) is 0 dB. The effect of the noise dominates the distribution in each case masking the spectral nature of the signals. IPS and the multi-voice *à trous* method did the best job suppressing the noise. For five voices, the signal is barely distinguishable and for ten voices the noise is sufficiently suppressed that the chirps become apparent.

6. UWB Radar Data For The Boat With Corner Reflector

Figure 40(a) presents the raw UWB radar signal return corresponding to the small boat with a corner reflector. The radar clearly detects the boat at bin 534.

The STFT, the Wigner-Ville distribution and IPS each clearly delineate the target transient, as shown in Figures 41-43. The frequency step and overlap was varied in the STFT and Wigner-Ville methods but no appreciable processing gain was achieved.

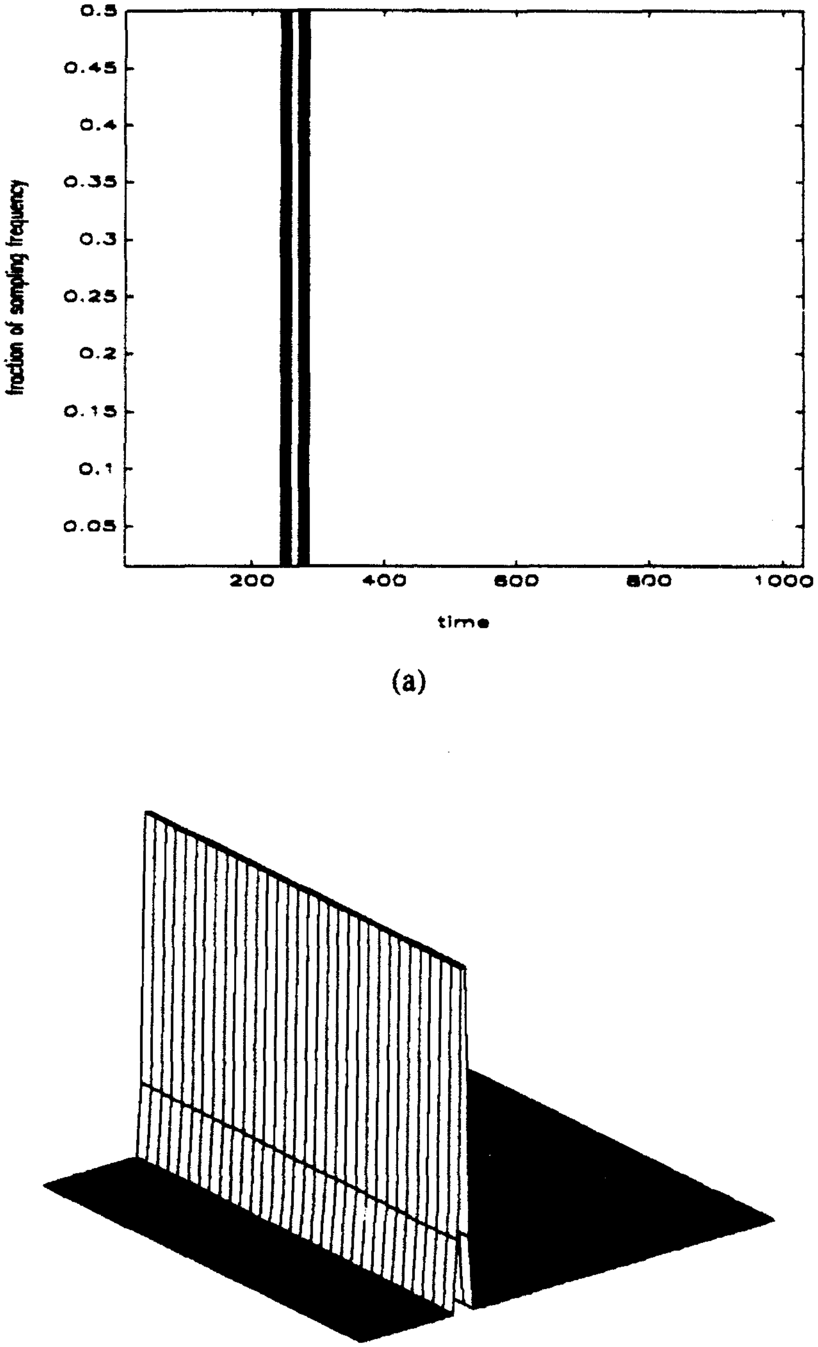
In Figures 44-47, the three DWT algorithms each display superior time resolution, however the one voice *à trous* provided the highest processing gain. Notice in Figure 46(b) that the transient is smoothed as the number of voices increases. This effect illustrates the tradeoff between time and frequency resolution.

7. UWB Radar Data For The Boat Without Corner Reflector

Figure 40(b) presents the raw UWB radar return signal corresponding to the small boat without the corner reflector. Figures 48-54 show the performance of each method applied to this signal. The STFT shown in Figure 48, the Wigner-Ville distribution shown in Figure 49 and Mallat's algorithm shown in Figure 54 are unable to distinguish the target in the presence of sea clutter and RFI. IPS is able to differentiate the target with a 128 point Hamming window and a four point step size. This implementation achieves a PGR of 0.0 dB, however, the target is visible as the small peak, offset with a slightly higher frequency than the main band. Although IPS discerns the target, this method is undesirable because $V_{boat}$is much lower than V_{noise}. Thus, this technique will result in a very low detection rate.

The DCWT detects the target with a processing gain of 24.59 dB and with excellent time resolution. The *à trous* algorithm, shown in Figure 52 provides best joint time-frequency resolution and the best of processing gain of 40.83 dB and both wavelet transforms suppressed the noise uniformly. For the DCWT and *à trous* algorithms a β

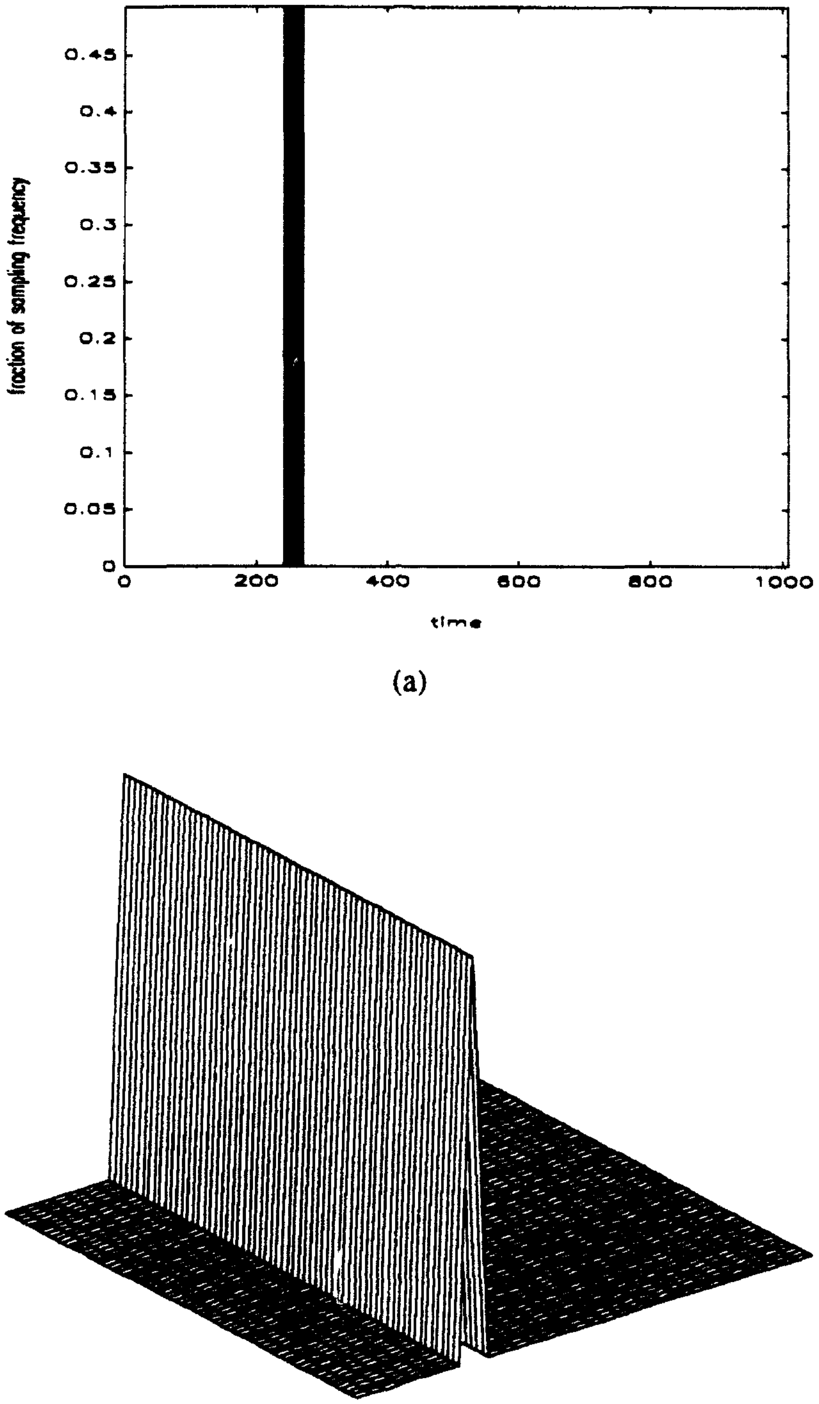
of 0.35 was used to achieve the high processing gains. The smaller β served to lower the relative bandwidth of the Morlet window and to provide a better match to the transient.



(a)

(b)

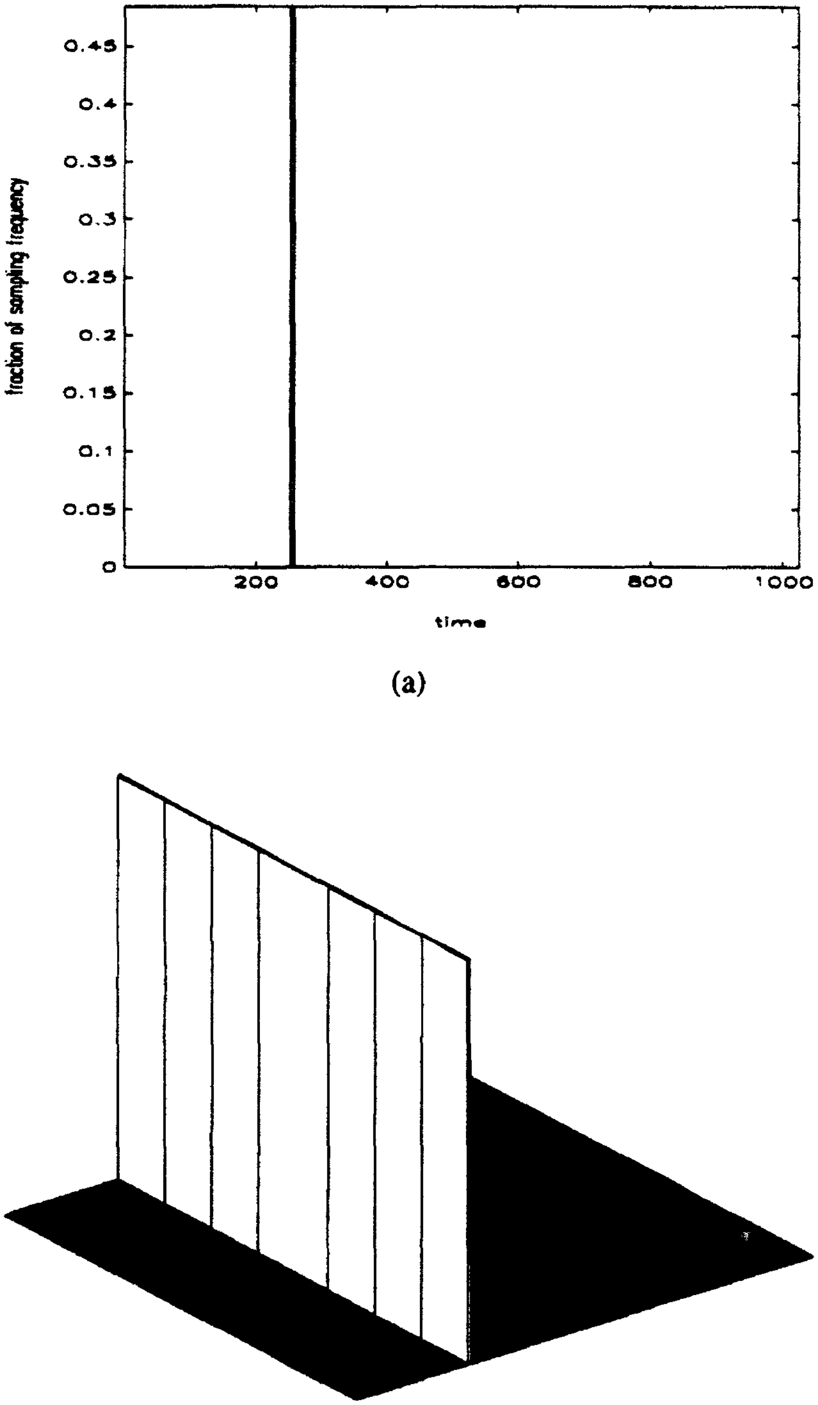
Figure 6. STFT time-frequency distribution for an impulse function at bin 256 (41 point Chebyshev window with a 10 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

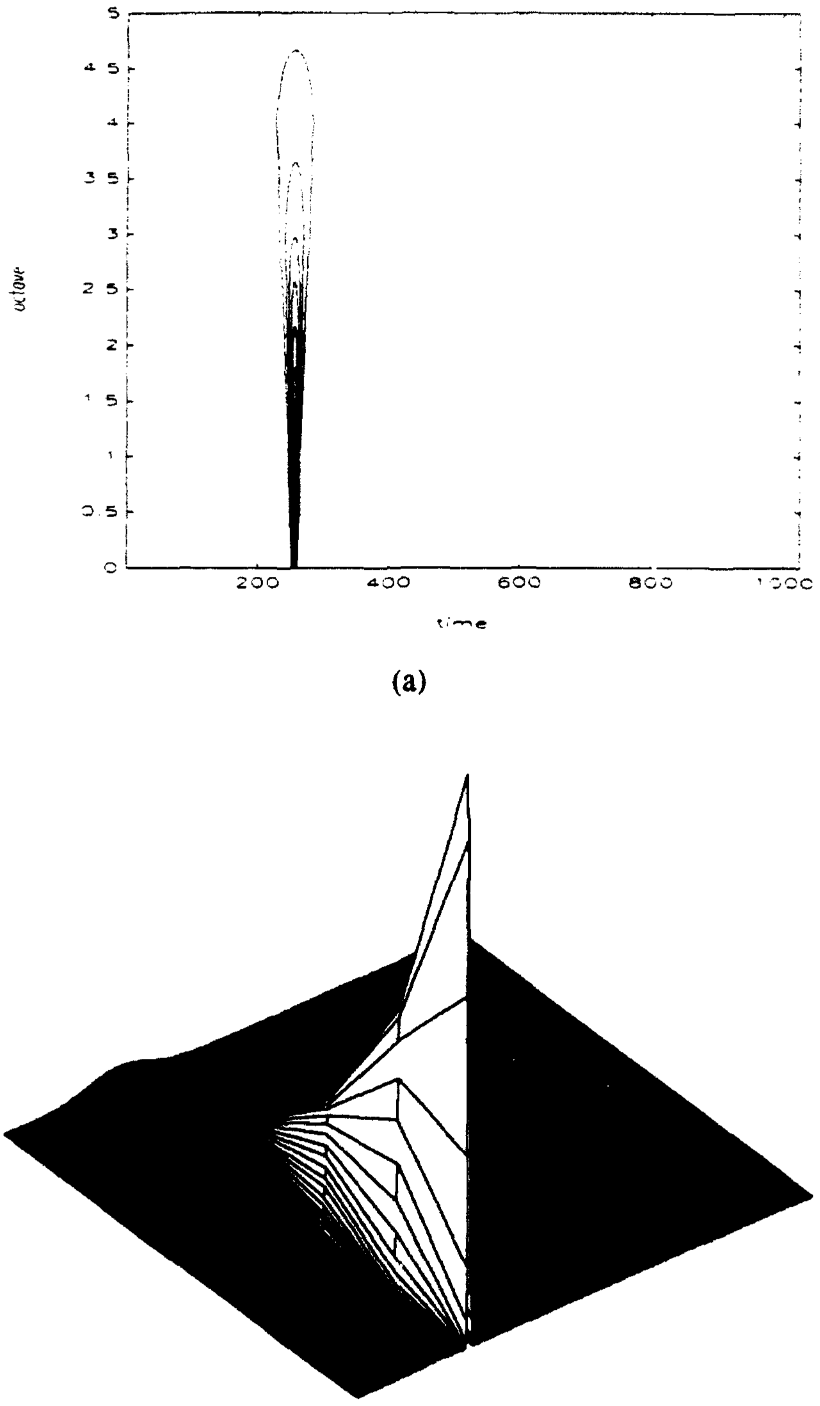
Figure 7. Wigner-Ville time-frequency distribution for an impulse at bin 256 (64 point window with a 32 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

Figure 8. IPS time-frequency distribution for an impulse function at bin 256 (64 point window with an 8 point step)
(a) Contour plot
(b) Mesh plot



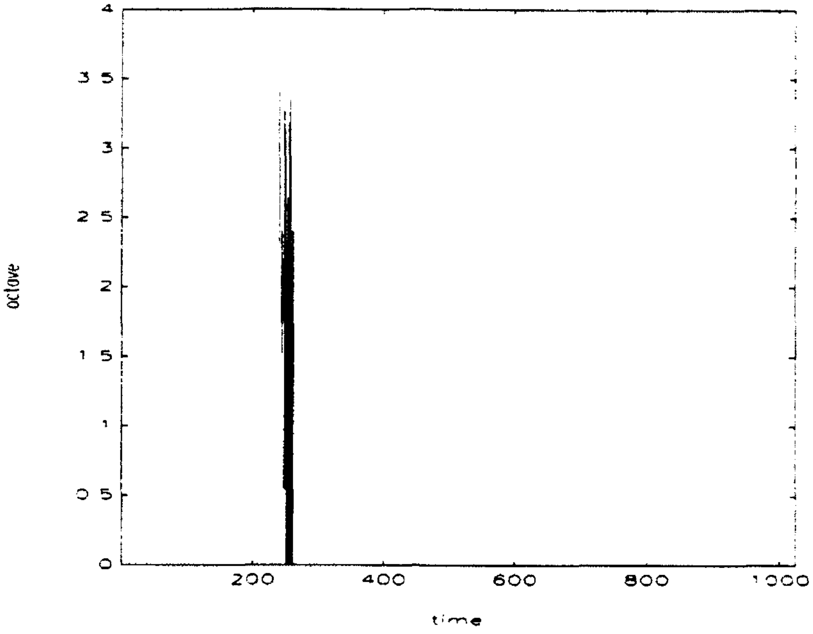
(a)

(b)

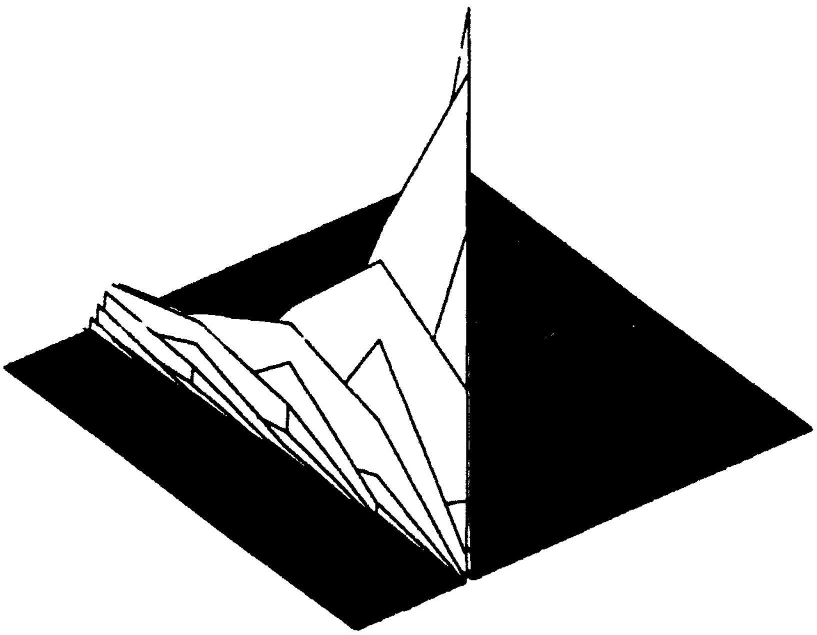
Figure 9. DCWT time-scale distribution for an impulse function at bin 256 (with k=π and β=0.6)

(a) Contour plot

(b) Mesh plot

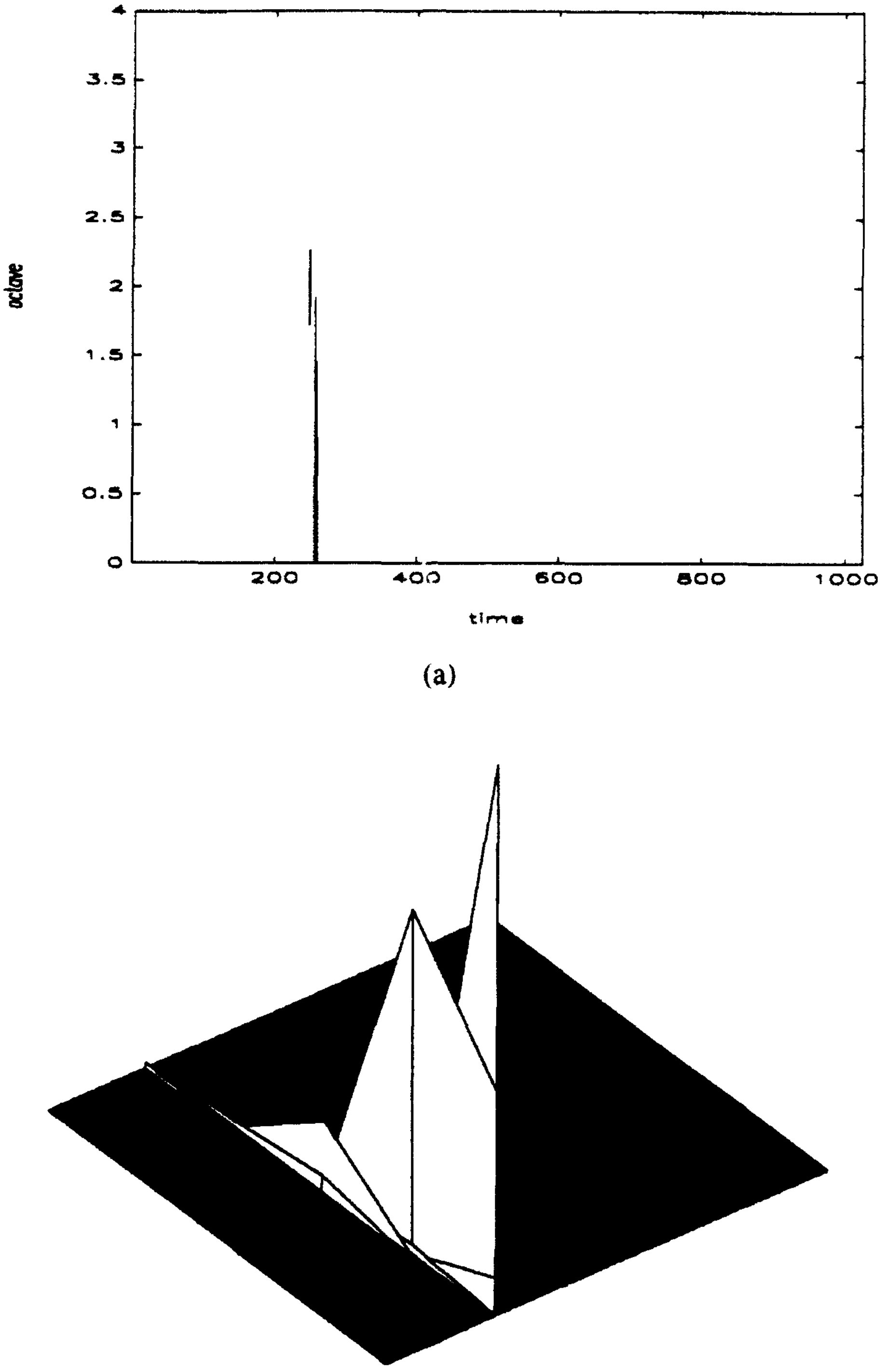


(a)



(b)

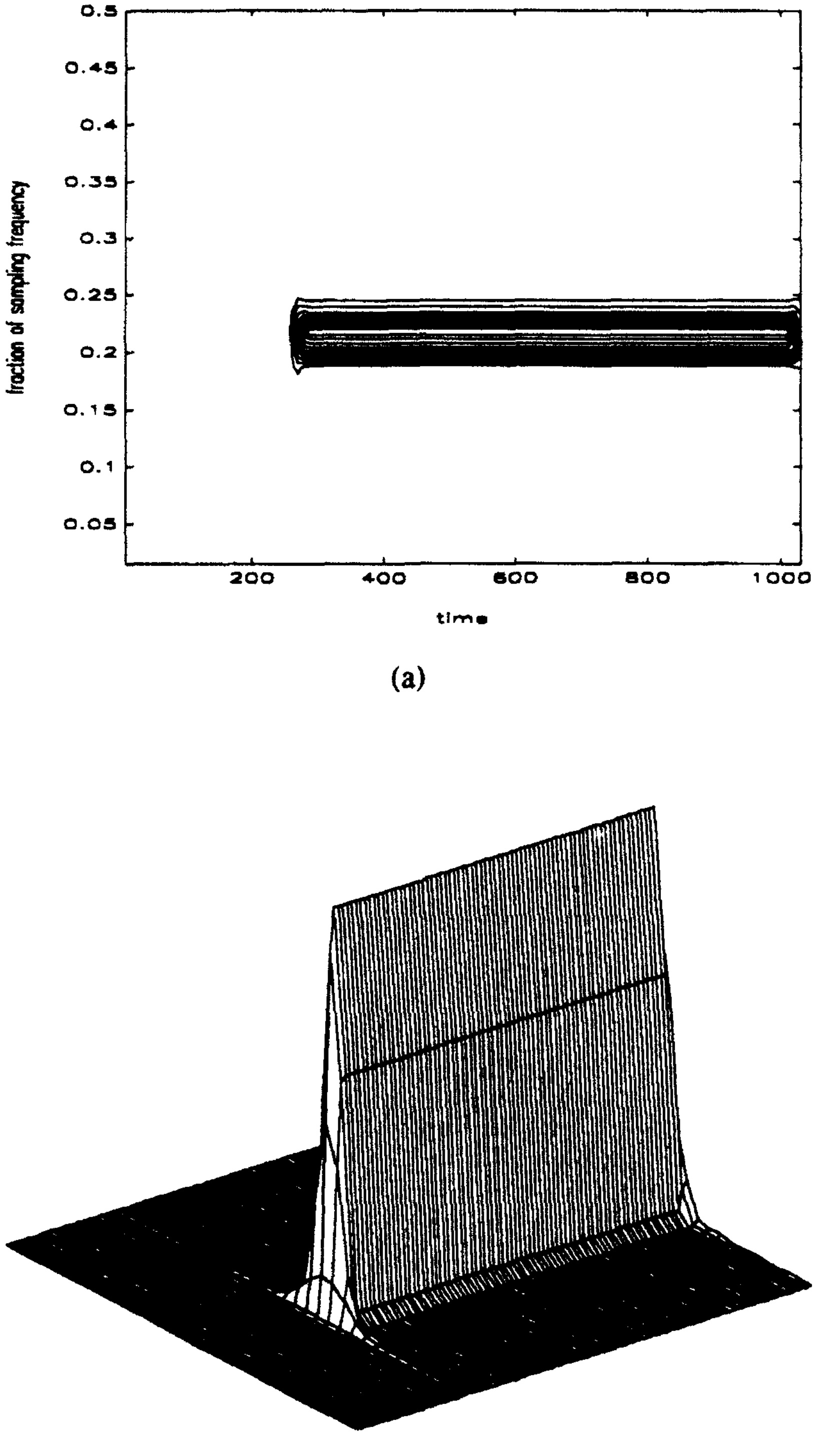
Figure 10. DWT (*à trous*) time-scale distribution for an impulse function at bin 256 (with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



(a)

(b)

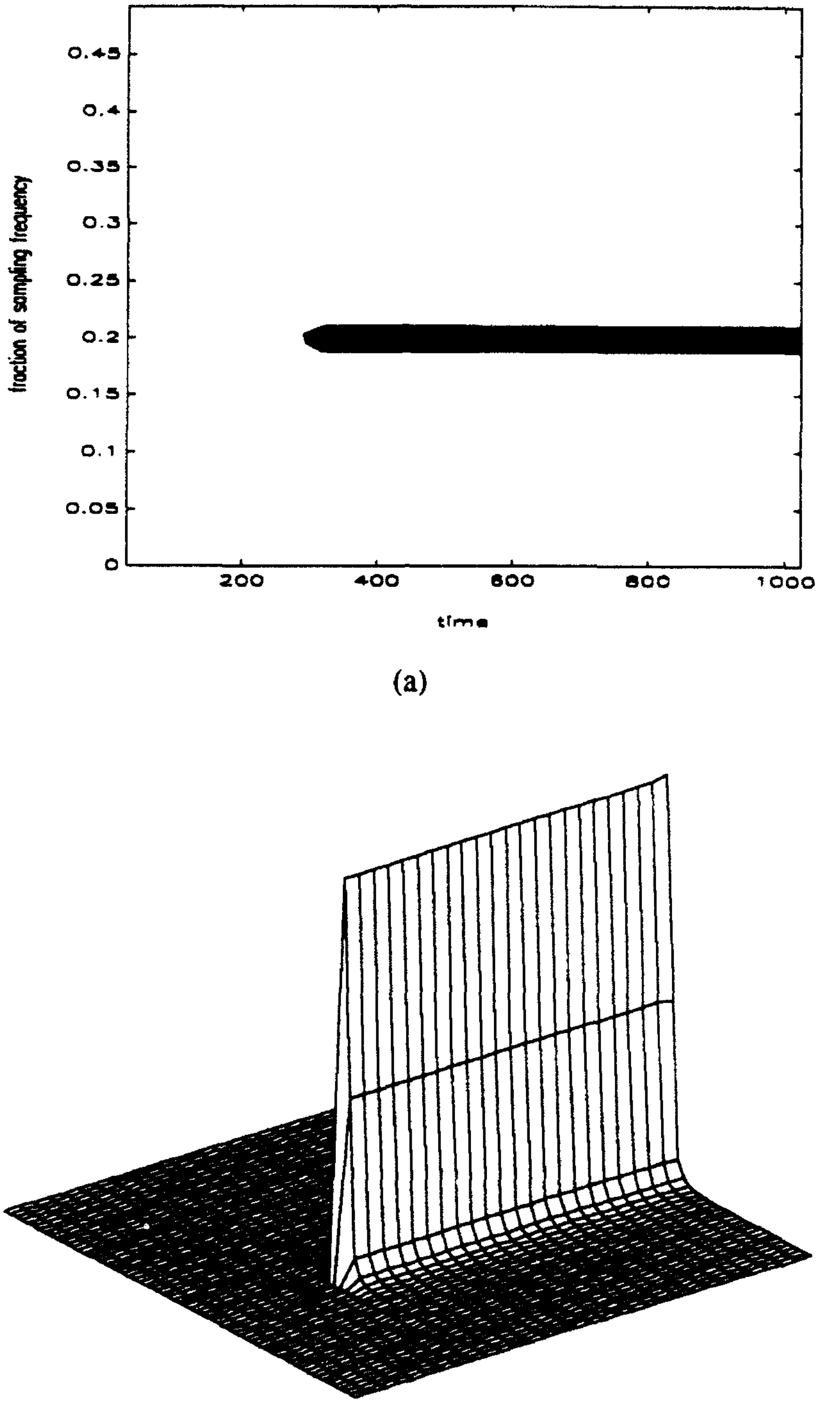
Figure 11. DWT (Mallat) time-scale distribution for an impulse function at bin 256 (12 point scaling function)
(a) Contour plot
(b) Mesh plot



(a)

(b)

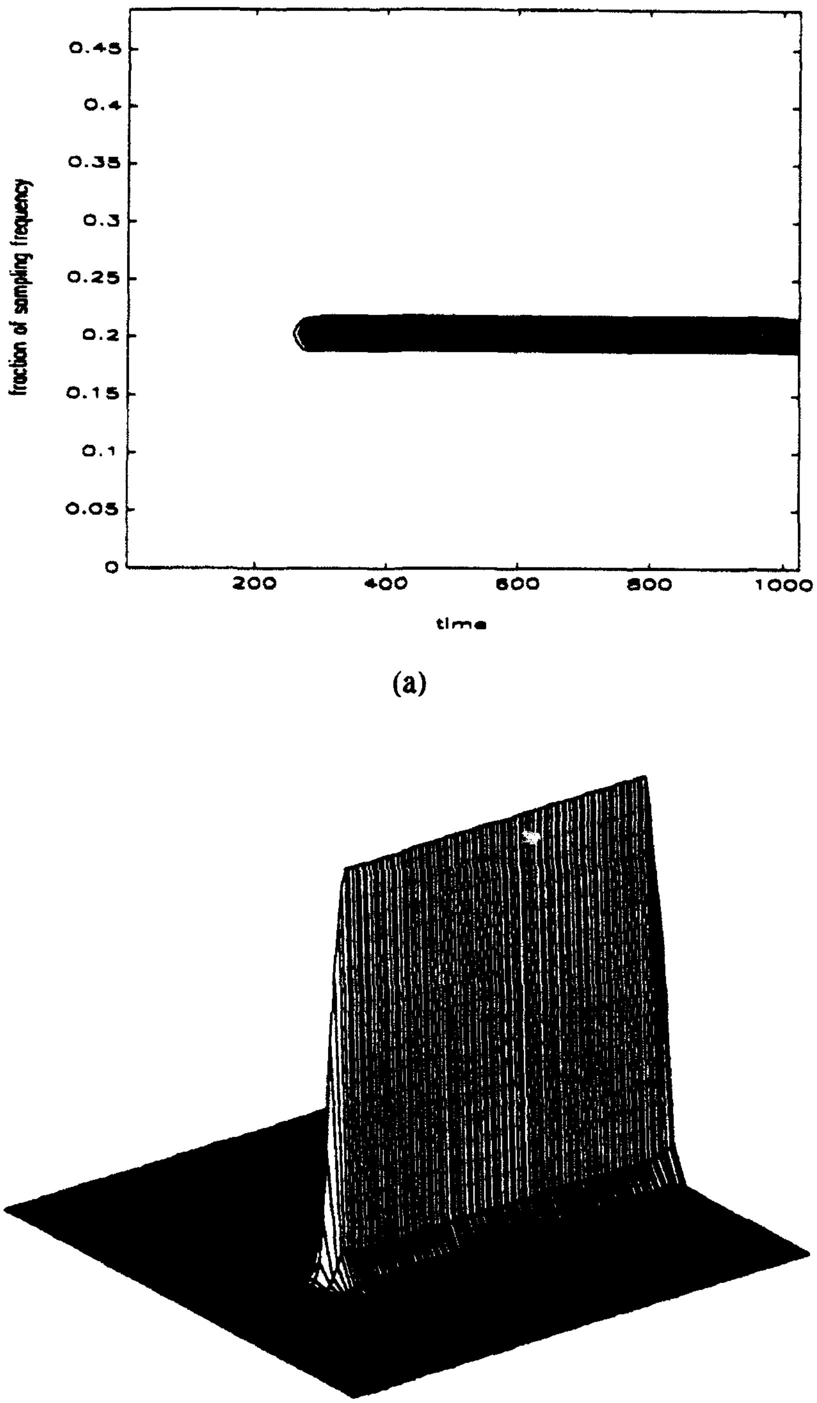
Figure 12. STFT time-frequency distribution for a complex sinusoid beginning at bin 256 (41 point Chebyshev window with a 10 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

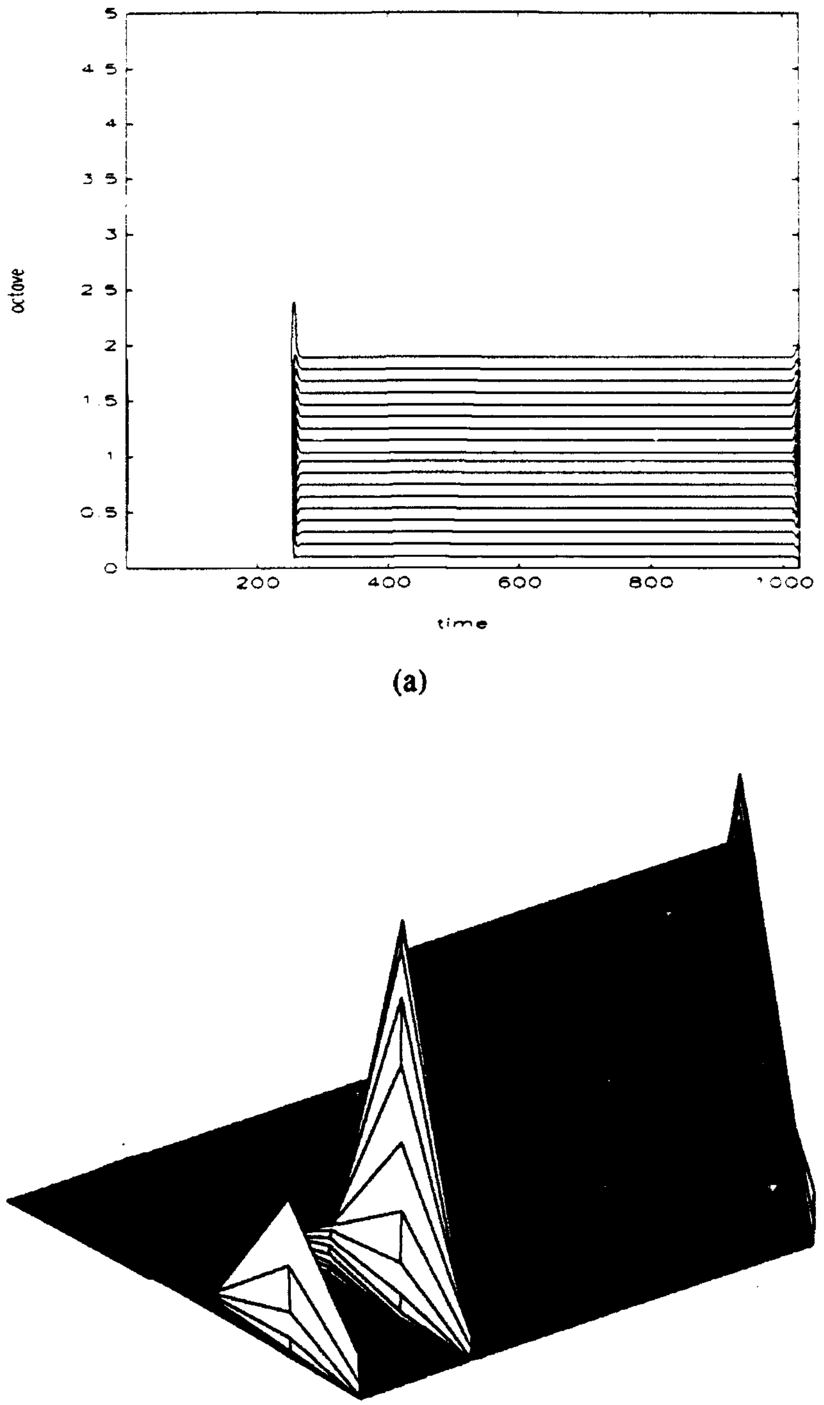
Figure 13. Wigner-Ville time-frequency distribution for a complex sinusoid beginning at bin 256 (64 point window with a 32 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

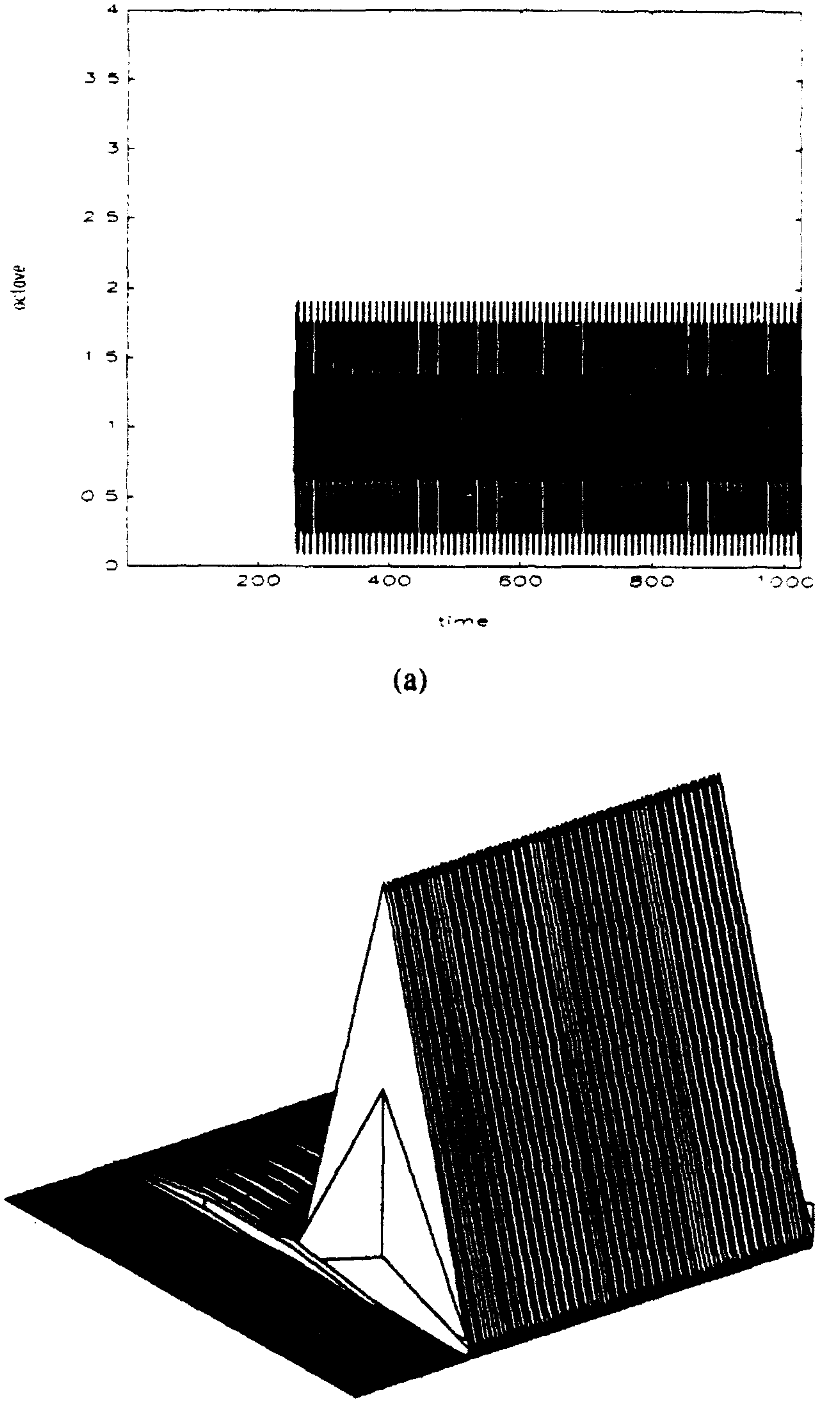
Figure 14. IPS time-frequency distribution for a complex sinusoid beginning at bin 256 (64 point window with an 8 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

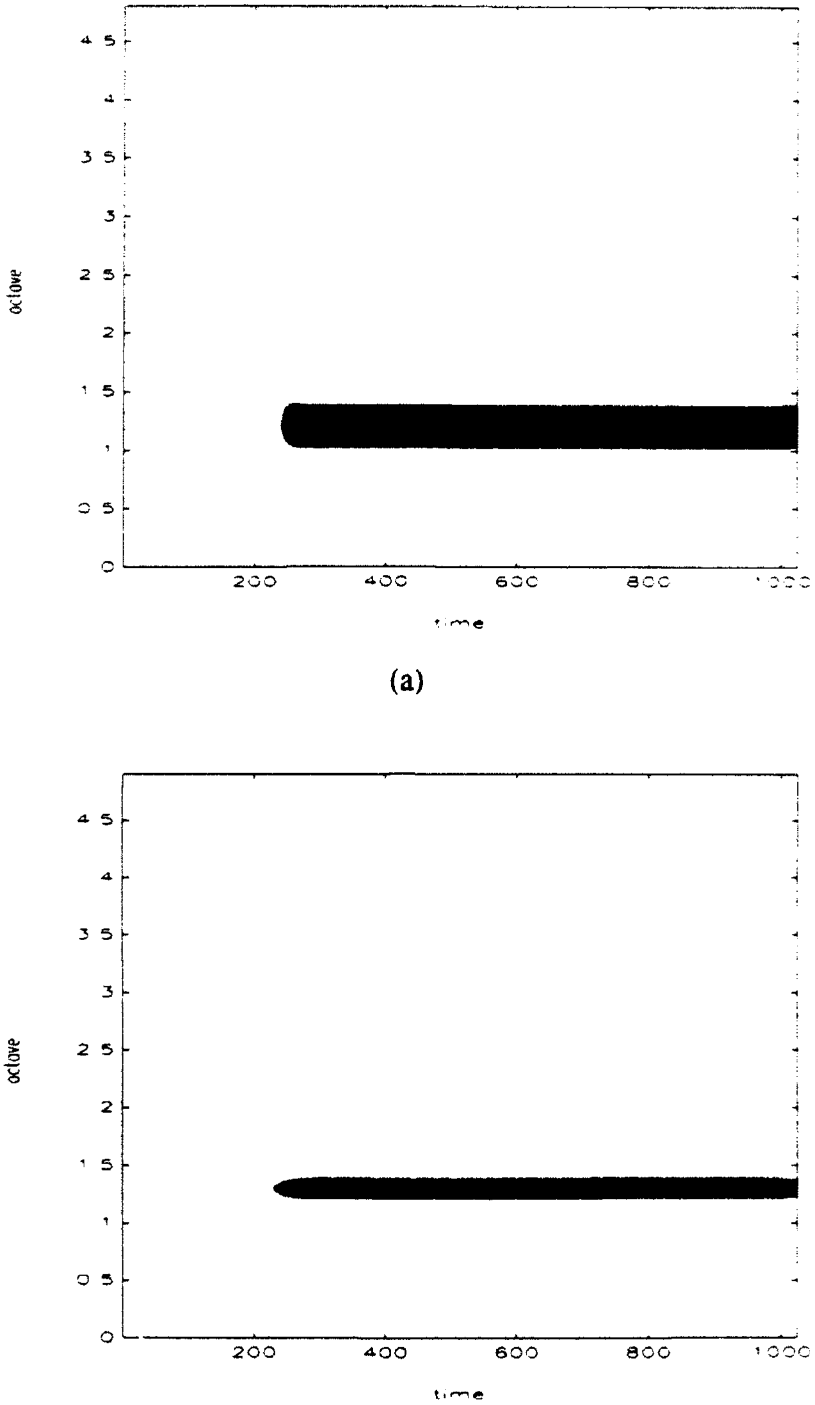
Figure 15. DCWT time-scale distribution for a complex sinusoid beginning at bin 256 (with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



(a)

(b)

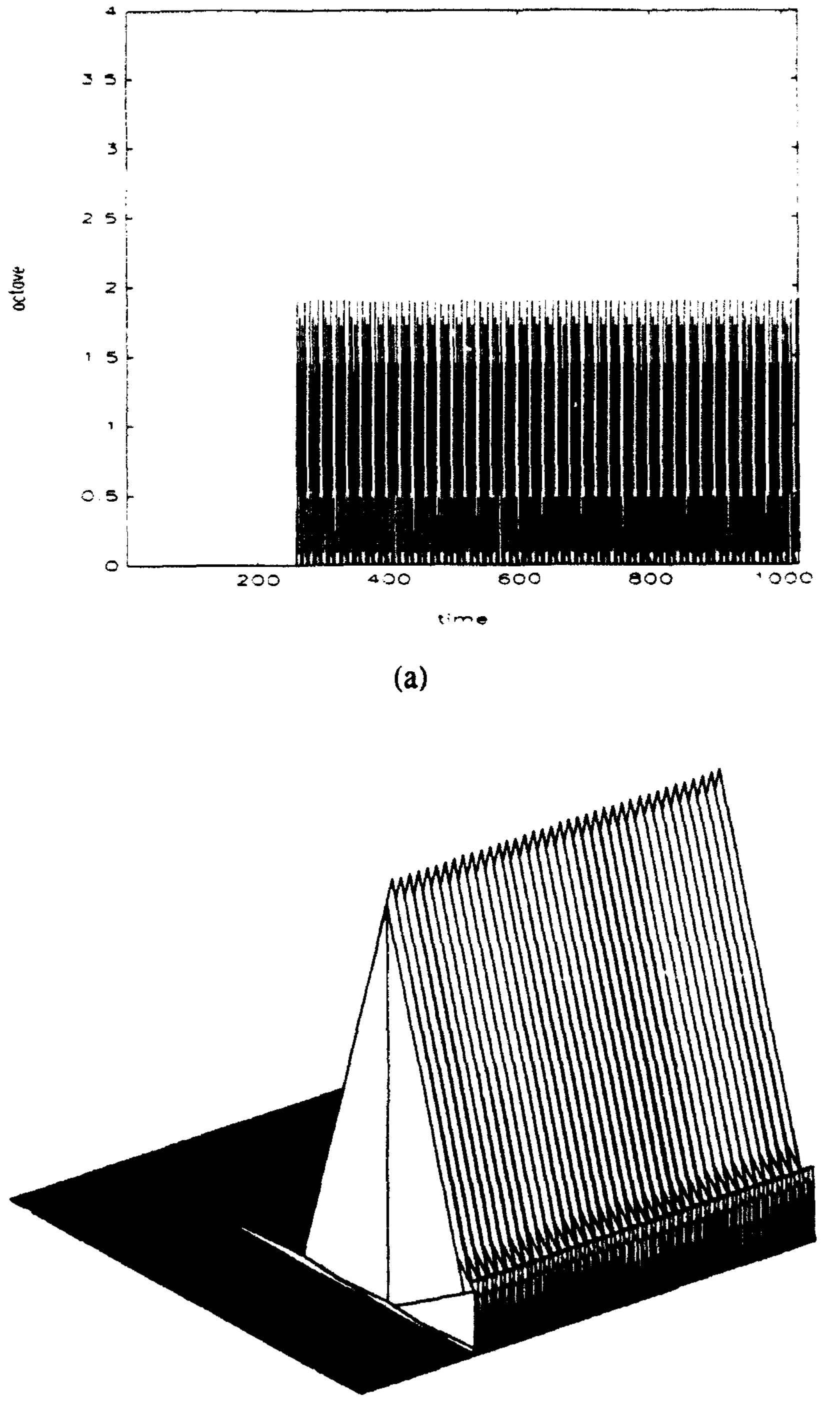
Figure 16. DWT (*à trous*) time-scale distribution for a complex sinusoid beginning at bin 256 (one voice with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



(a)

(b)

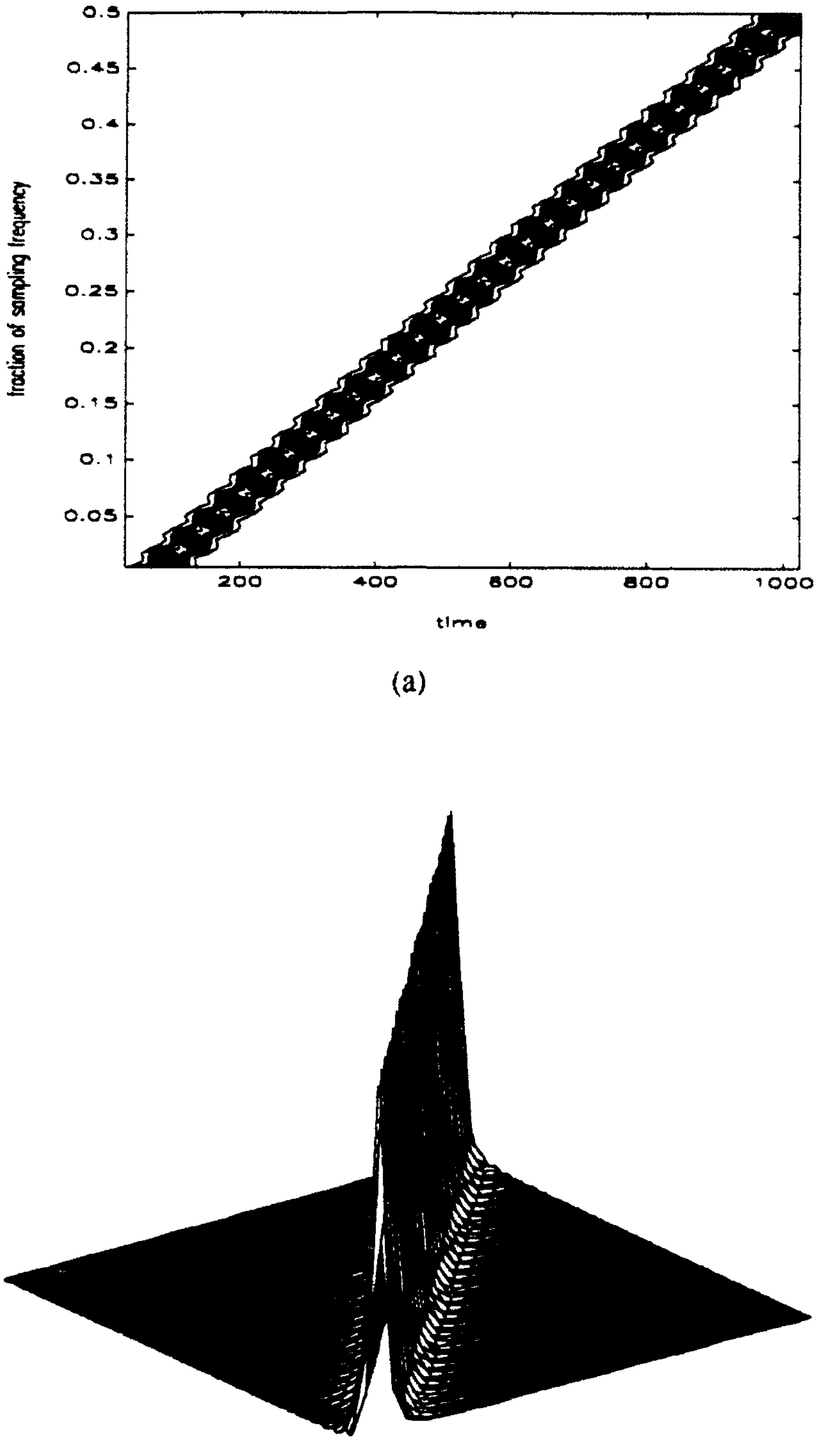
Figure 17. DWT (*à trous*) time-scale distribution for a complex sinusoid beginning at bin 256 (with k=π and β=0.6)
(a) Contour plot for five voices
(b) Contour plot of ten voices



(a)

(b)

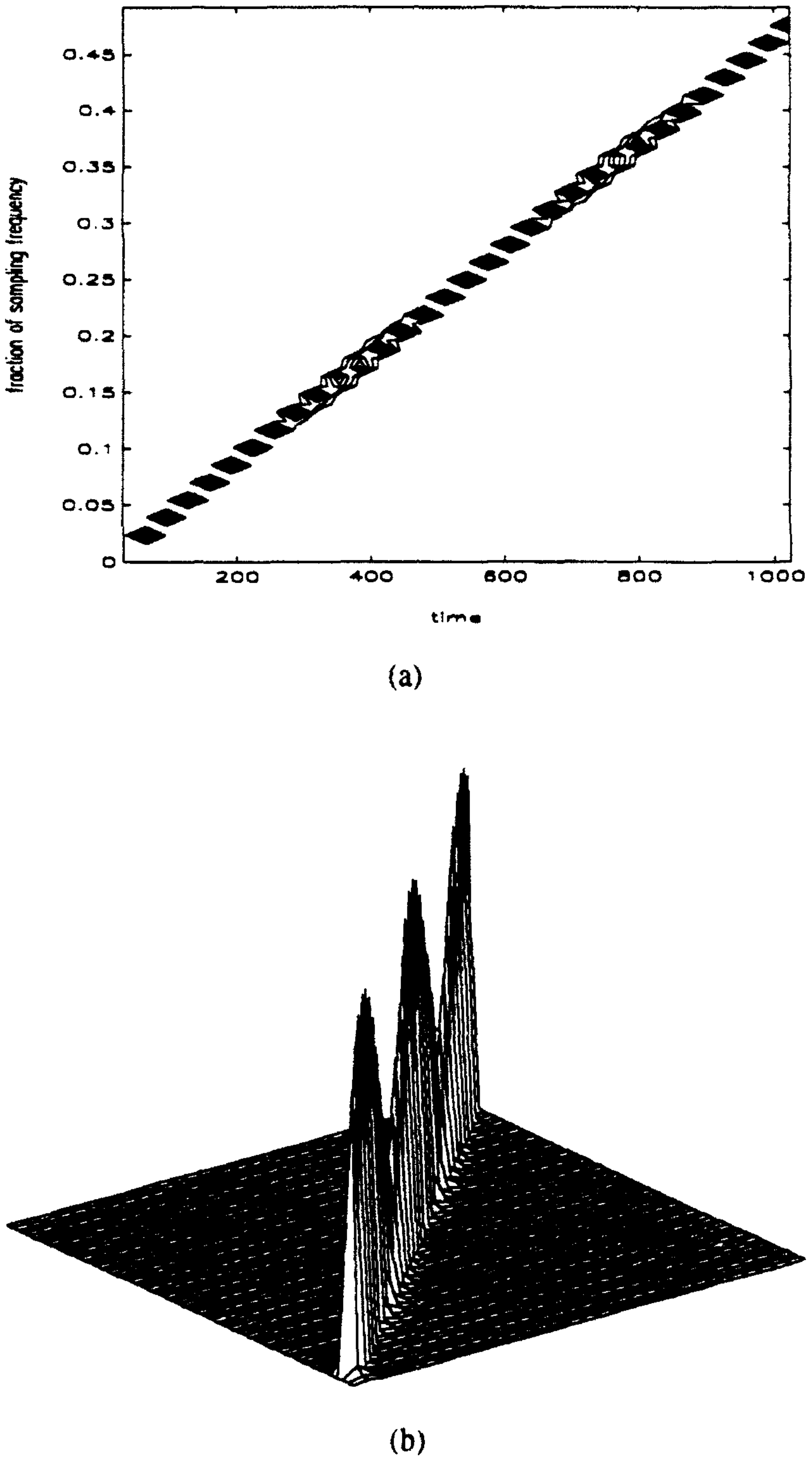
Figure 18. DWT (Mallat) time-scale distribution for a complex sinusoid beginning at bin 256 (12 point scaling function)
(a) Contour plot
(b) Mesh plot



(a)

(b)

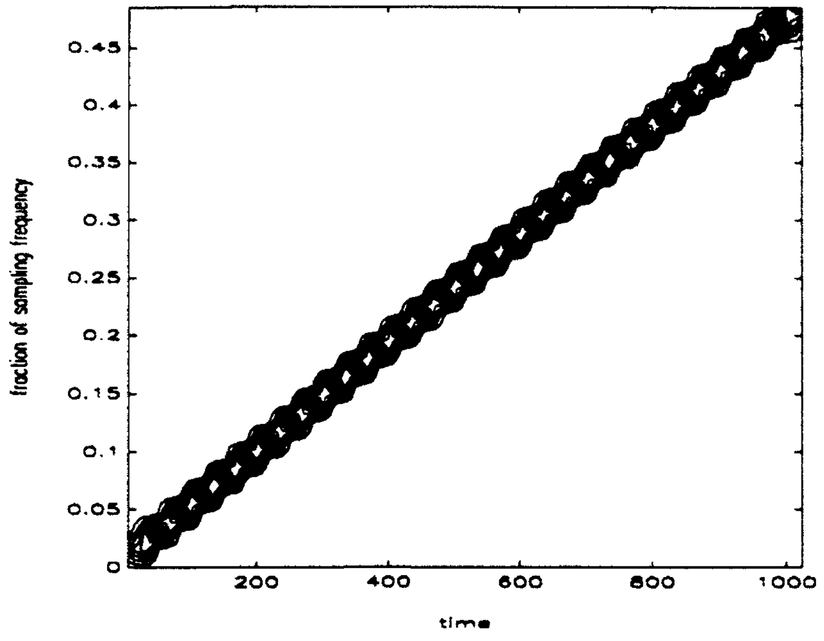
Figure 19. STFT time-frequency distribution for a linear chirp
(41 point Chebyshev window with a 10 point step)
(a) Contour plot
(b) Mesh plot



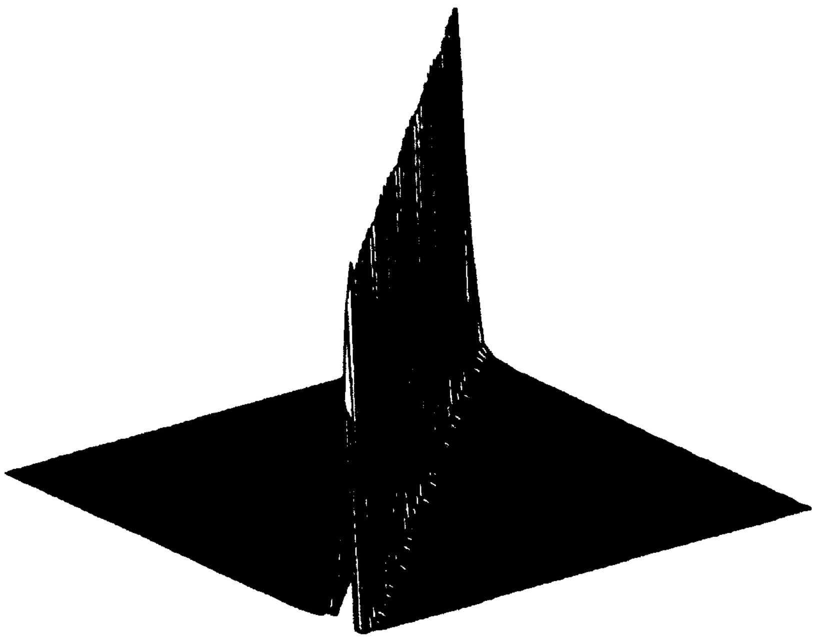
(a)

(b)

Figure 20. Wigner-Ville time-frequency distribution for a linear chirp
(64 point window with a 32 point step)
(a) Contour plot
(b) Mesh plot

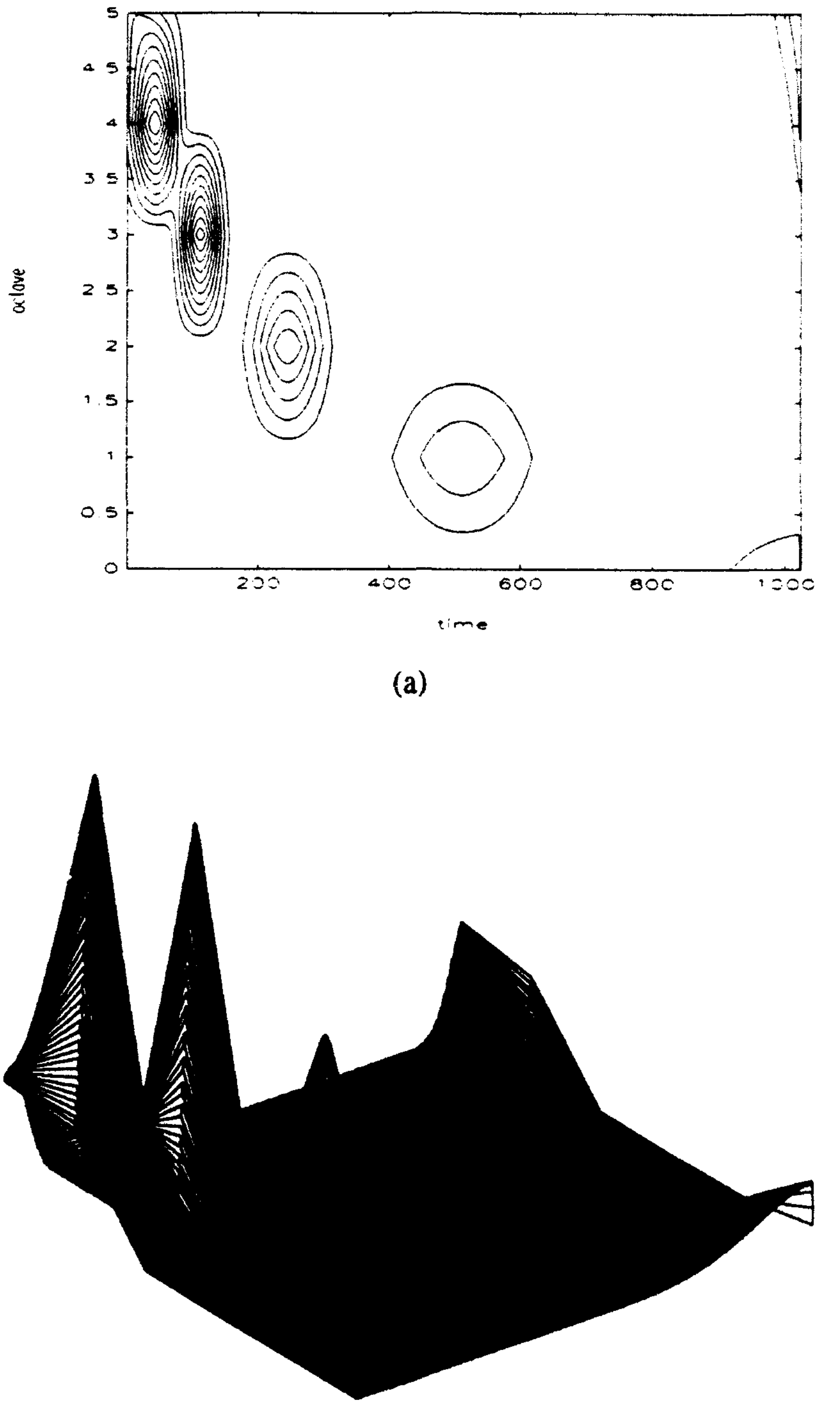


(a)



(b)

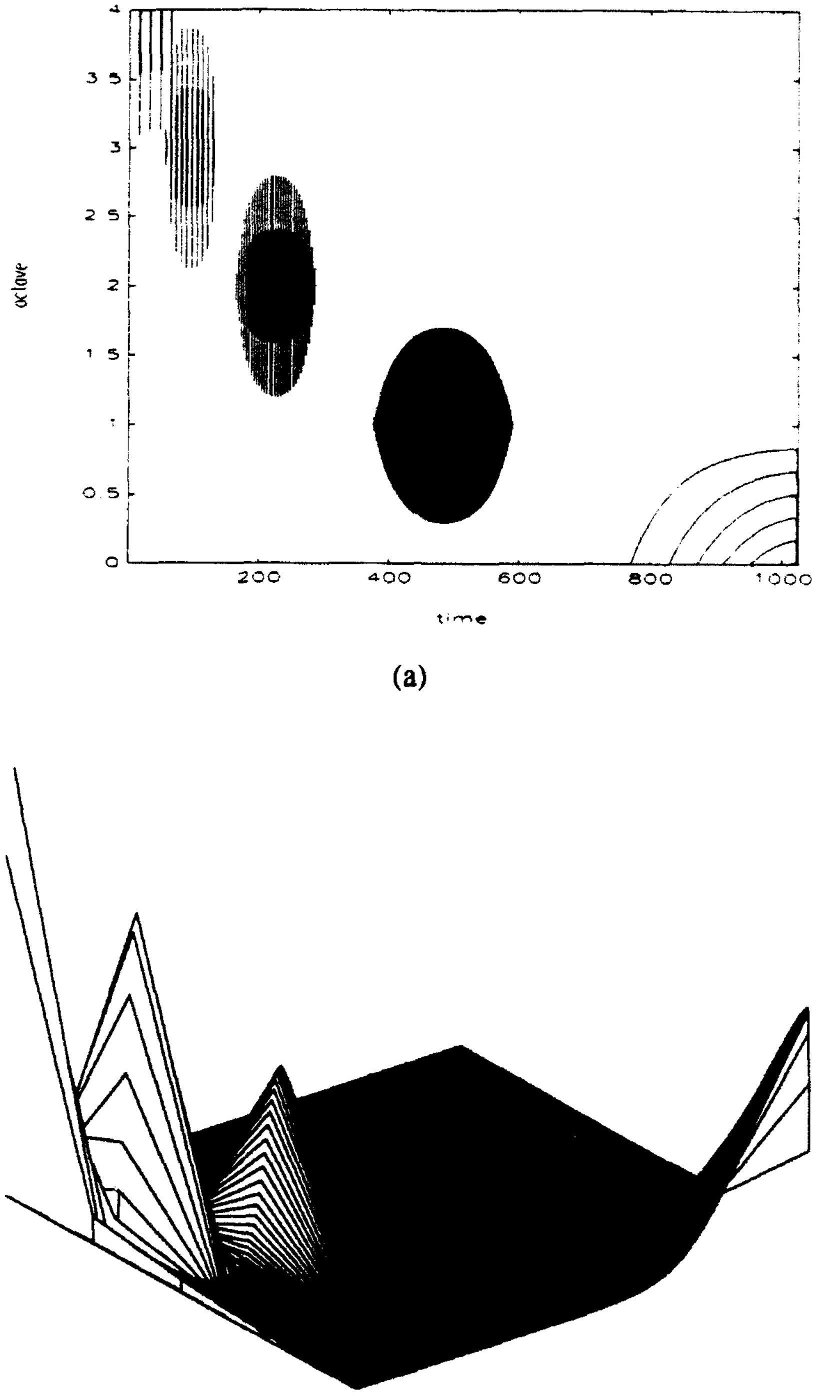
Figure 21. IPS time-frequency distribution for a linear chirp
(64 point window with an 8 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

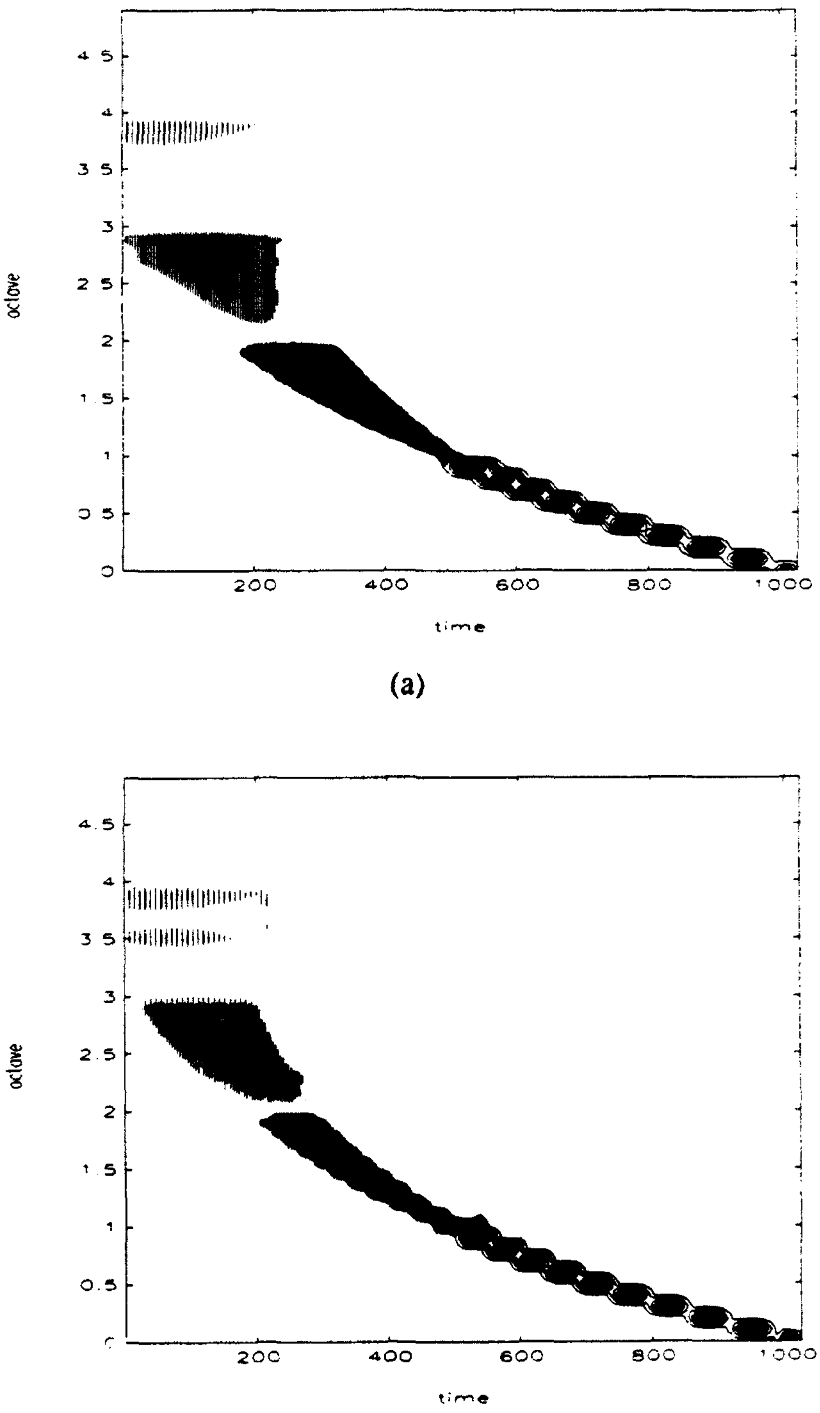
Figure 22. DCWT time-scale distribution for a linear chirp
(with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



(a)

(b)

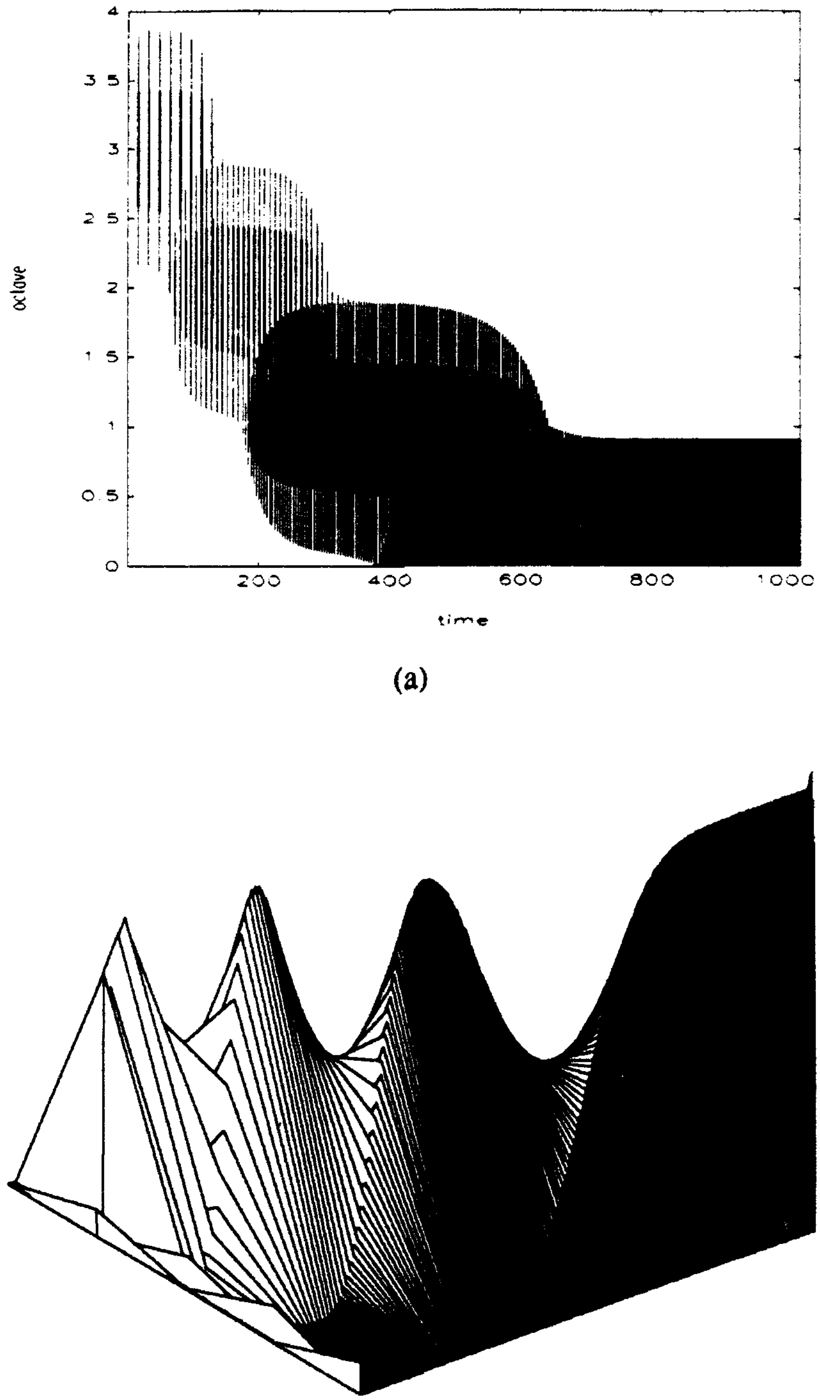
Figure 23. DWT (*à trous*) time-scale distribution for a linear chirp
(one voice with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



(a)

(b)

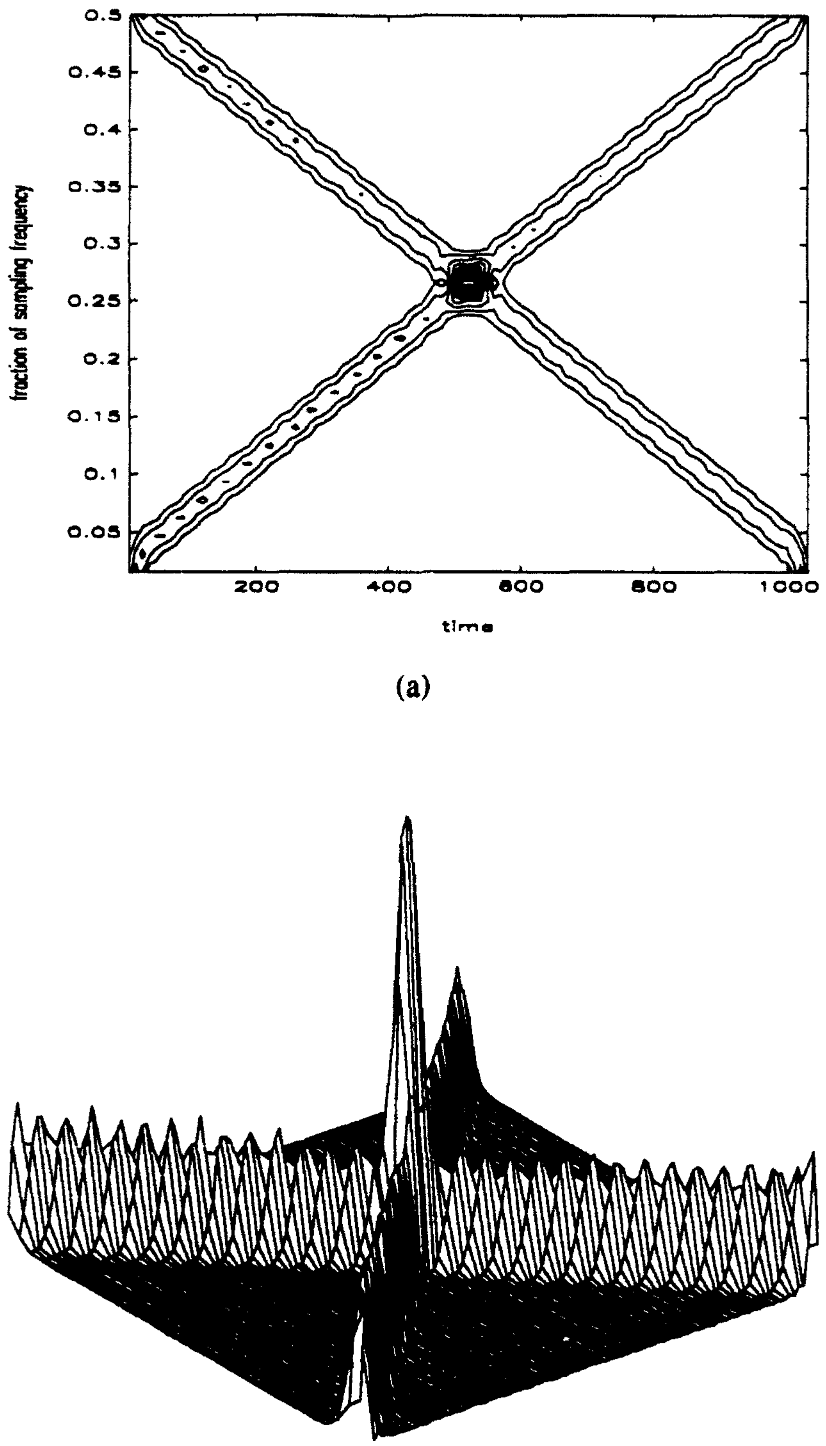
Figure 24. DWT (*à trous*) time-scale distribution for a linear chirp (with k=π and β=0.6)
(a) Contour plot for five voices
(b) Contour plot for ten voices



(a)

(b)

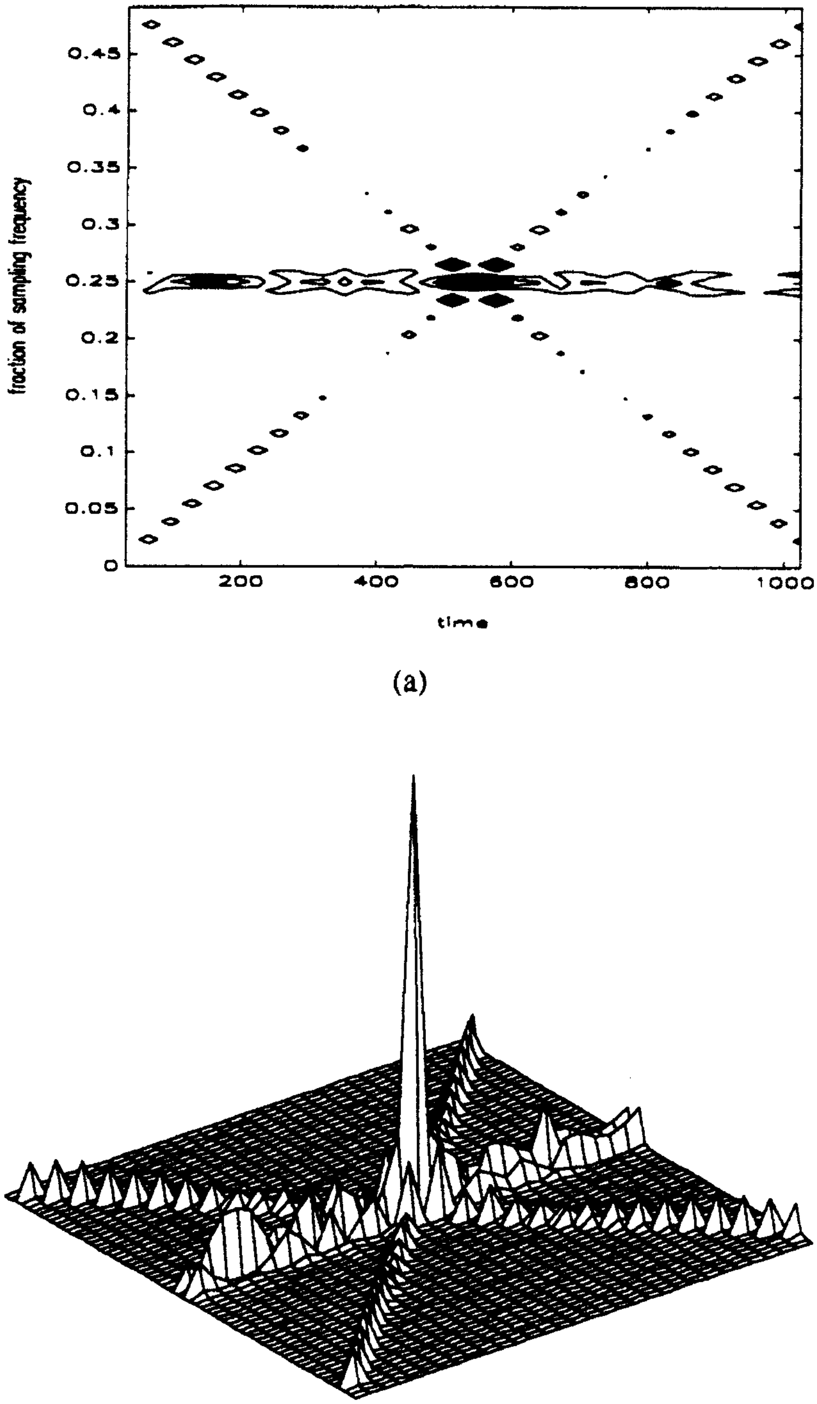
Figure 25. DWT (Mallat) time-scale distribution for a linear chirp
(12 point scaling function)
(a) Contour plot
(b) Mesh plot



(a)

(b)

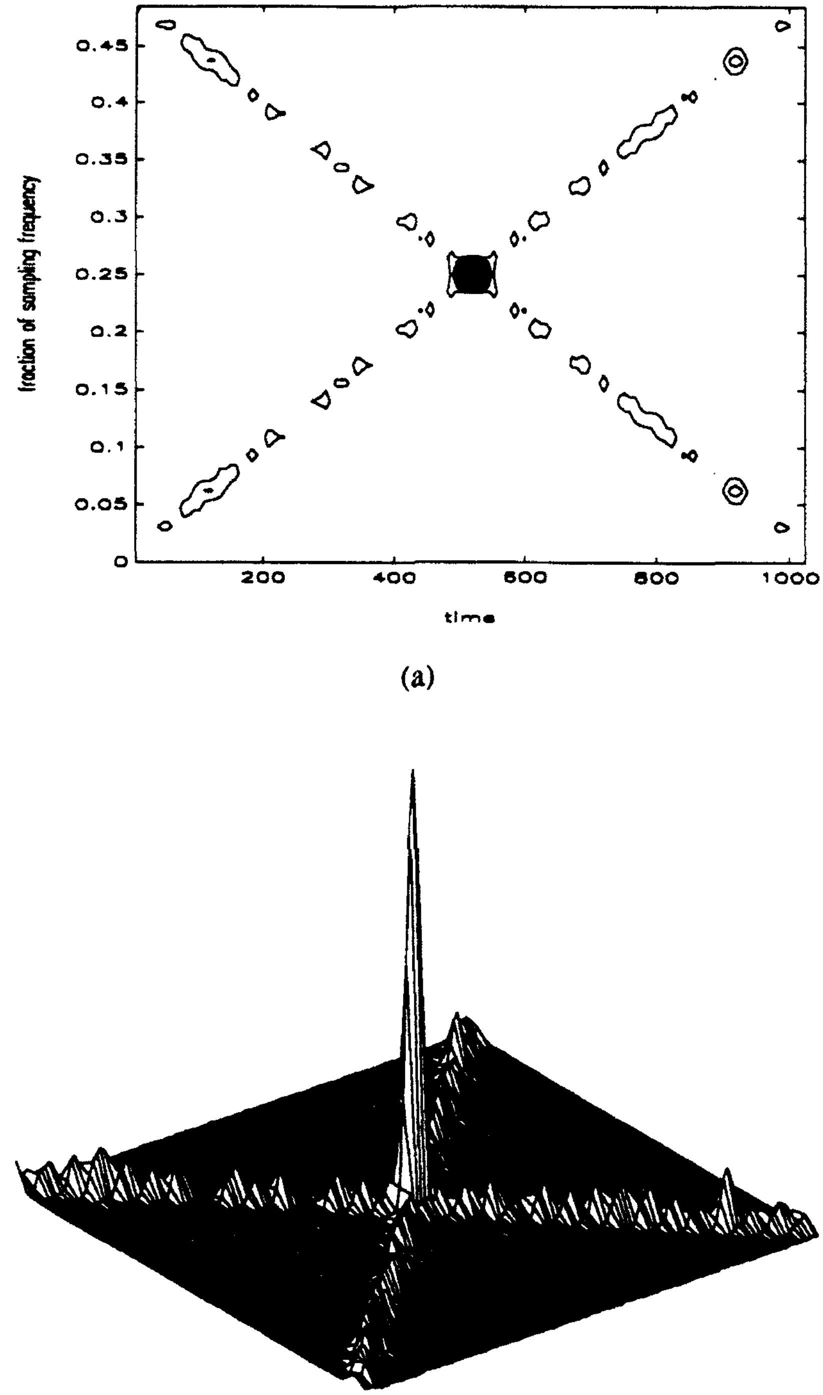
Figure 26. STFT time-frequency distribution for two linear chirps
(41 point Chebyshev window with a 10 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

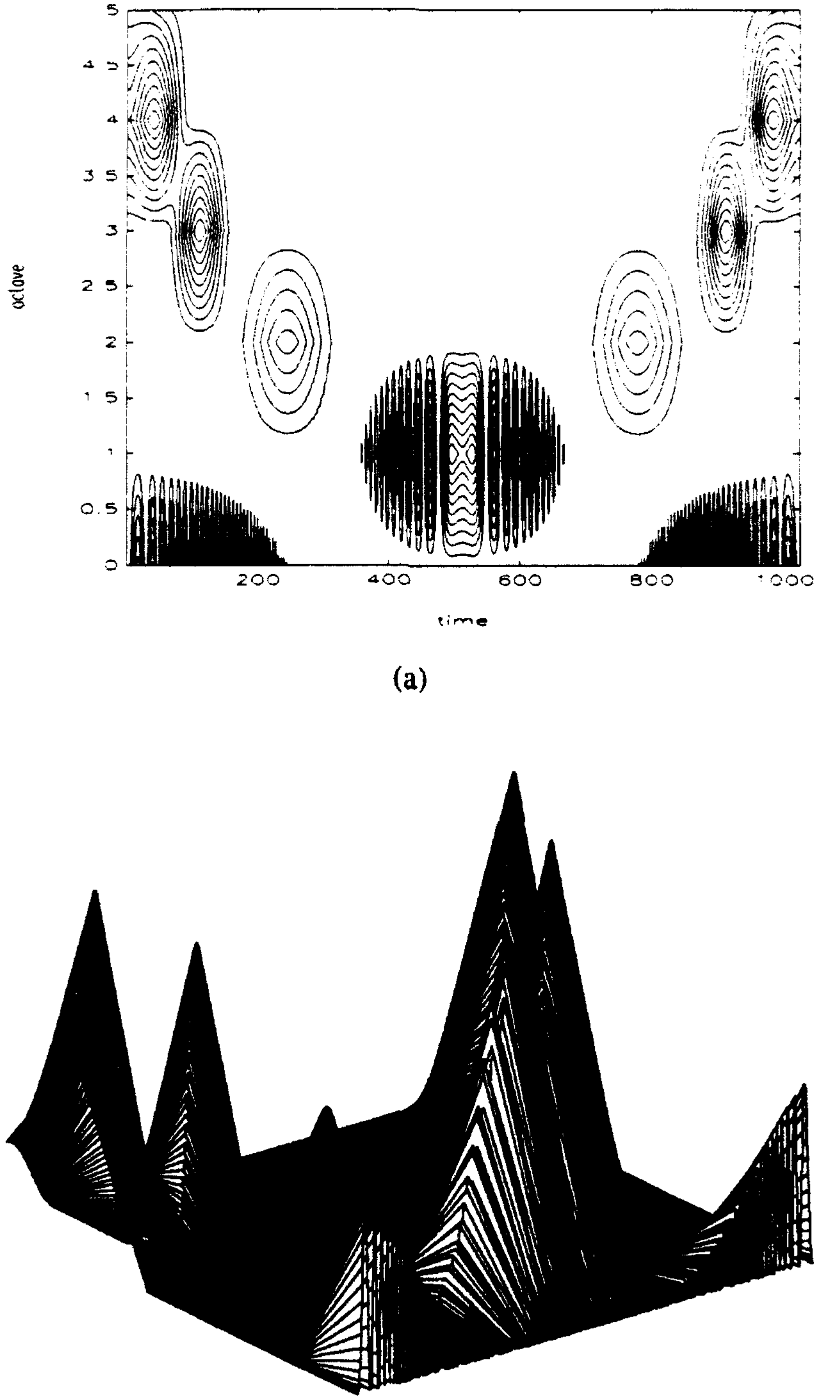
Figure 27. Wigner-Ville time-frequency distribution for two linear chirps
(64 point window with a 32 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

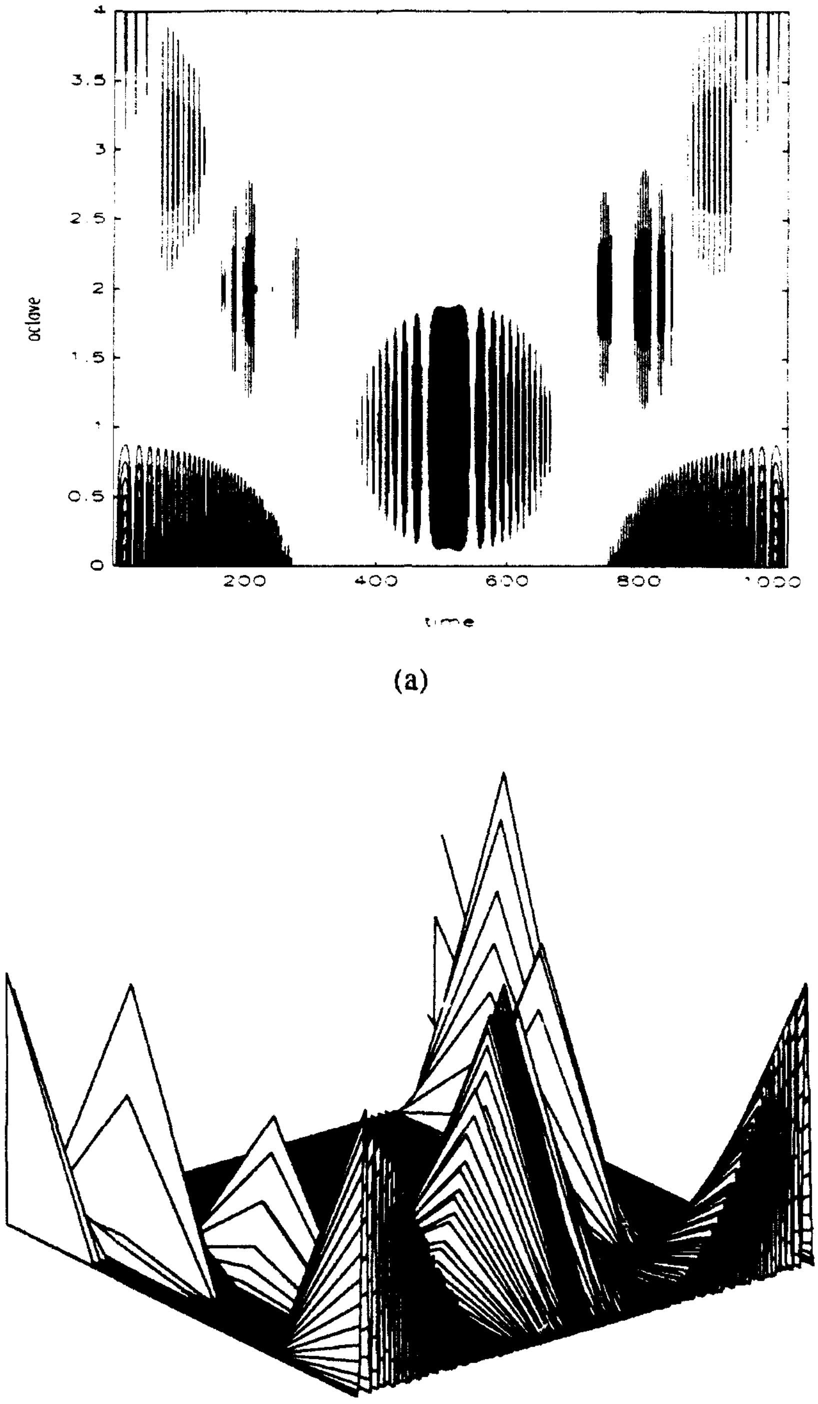
Figure 28. IPS time-frequency distribution for two linear chirps (64 point window with an 8 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

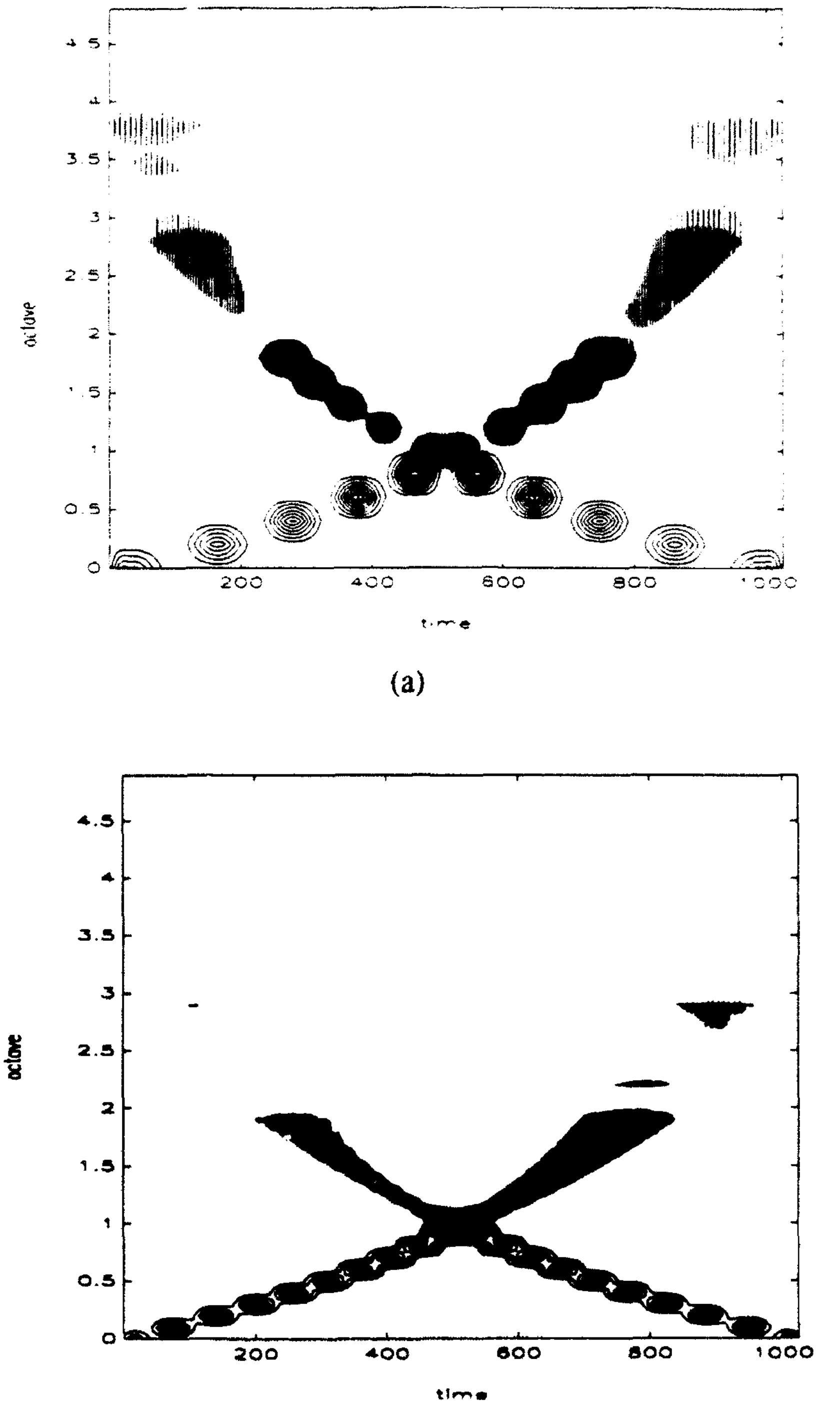
Figure 29. DCWT time-scale distribution for two linear chirps
(with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



(a)

(b)

Figure 30. DWT (*à trous*) time-scale distribution for two linear chirps (one voice with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



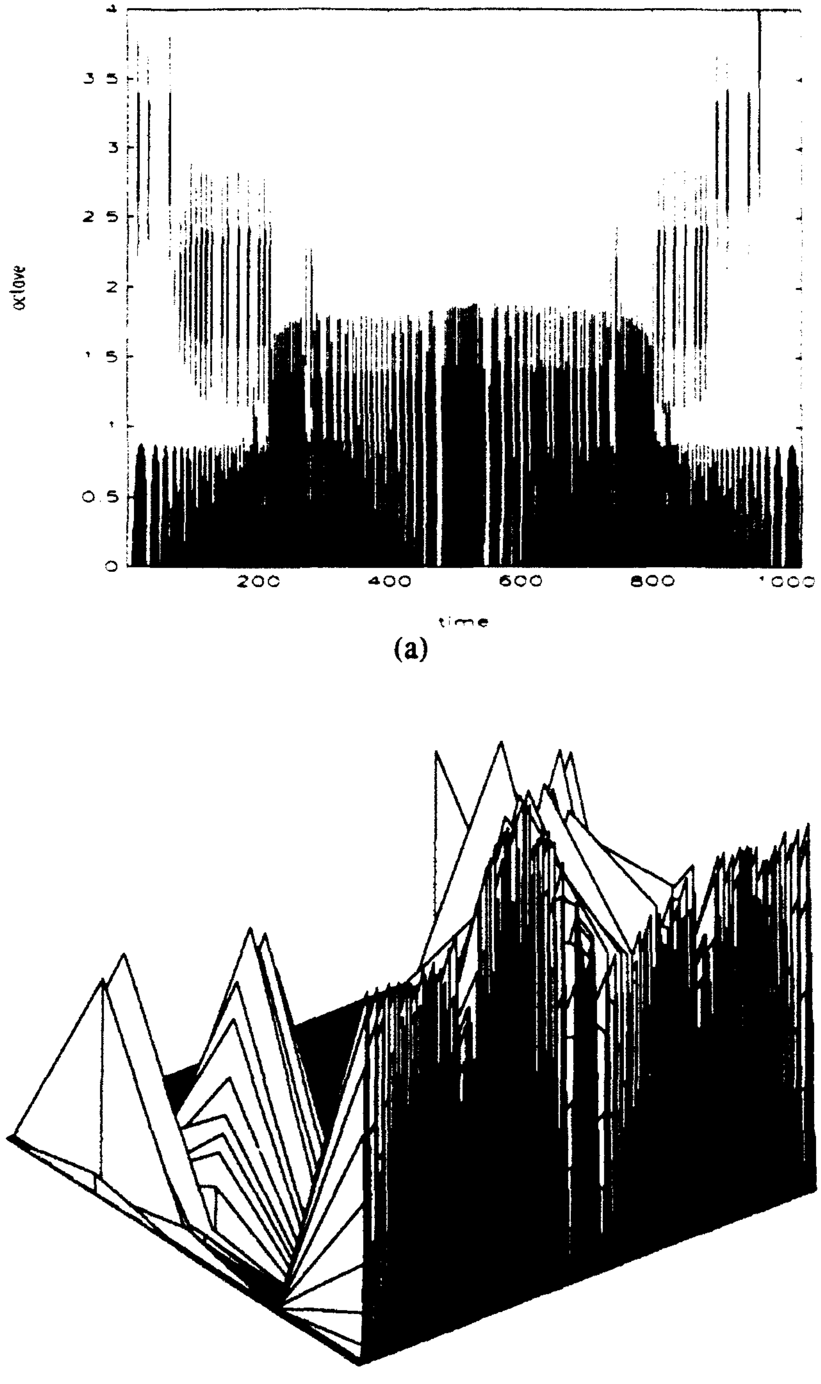
(a)

(b)

Figure 31. DWT (*a trous*) time-scale distribution for two linear chirps (with k=π and β=0.6)

(a) Contour plot for five voices

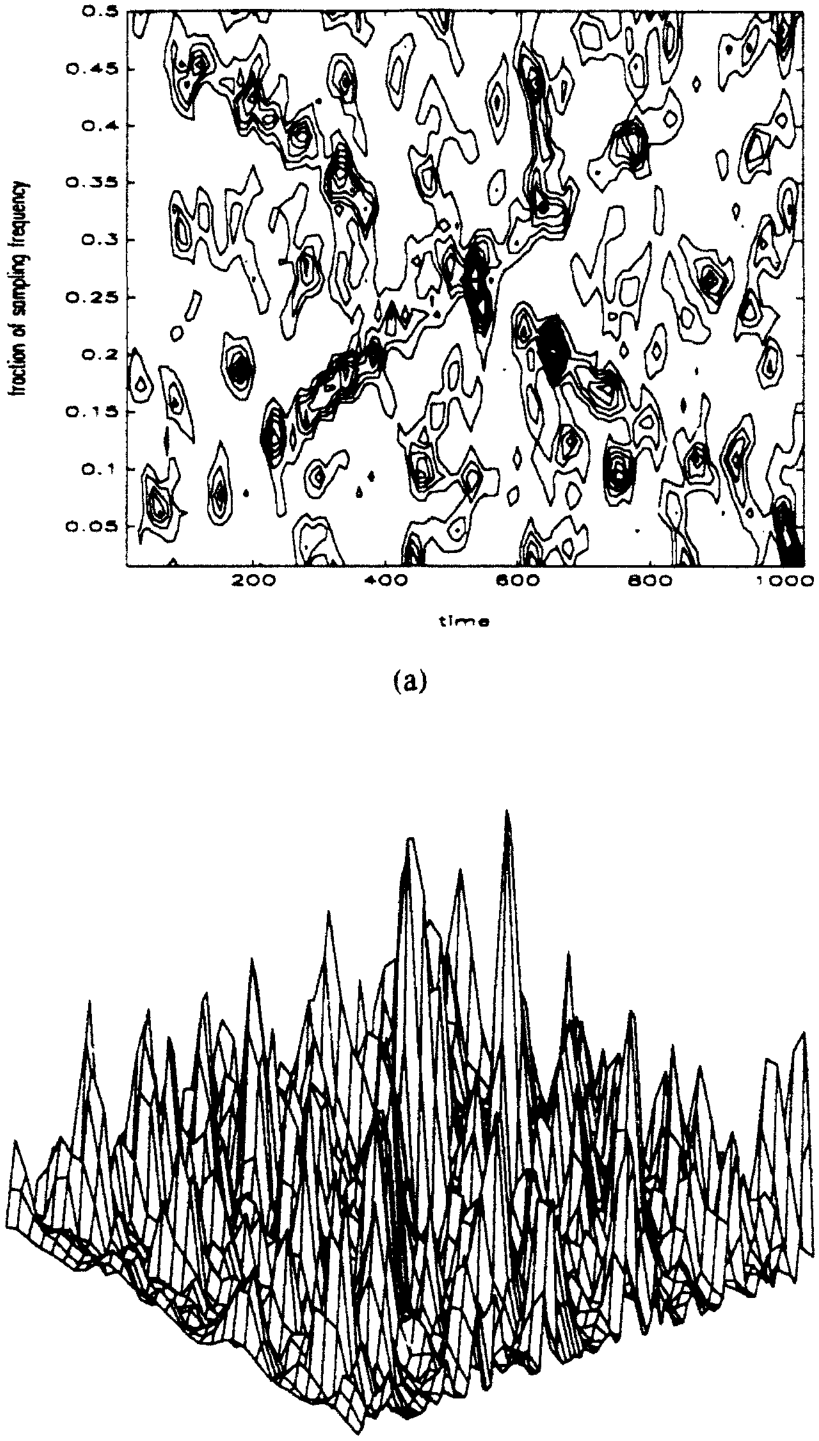
(b) Contour plot for ten voices



(a)

(b)

Figure 32. DWT (Mallat) time-scale distribution for two linear chirps
(12 point scaling function)
(a) Contour plot
(b) Mesh plot



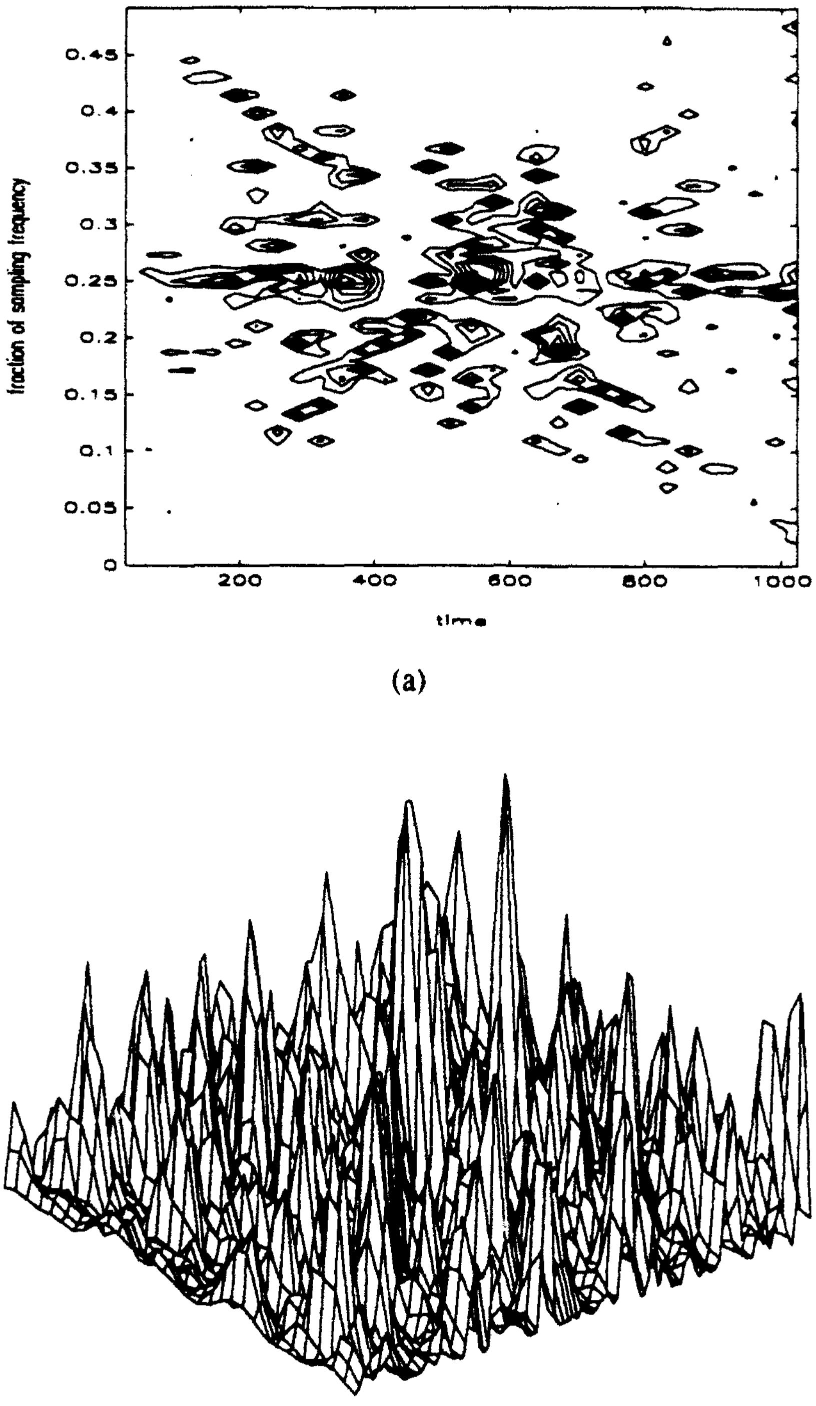
(a)

(b)

Figure 33. STFT time-frequency distribution for two linear chirps in noise (41 point Chebyshev window with a 10 point step)

(a) Contour plot

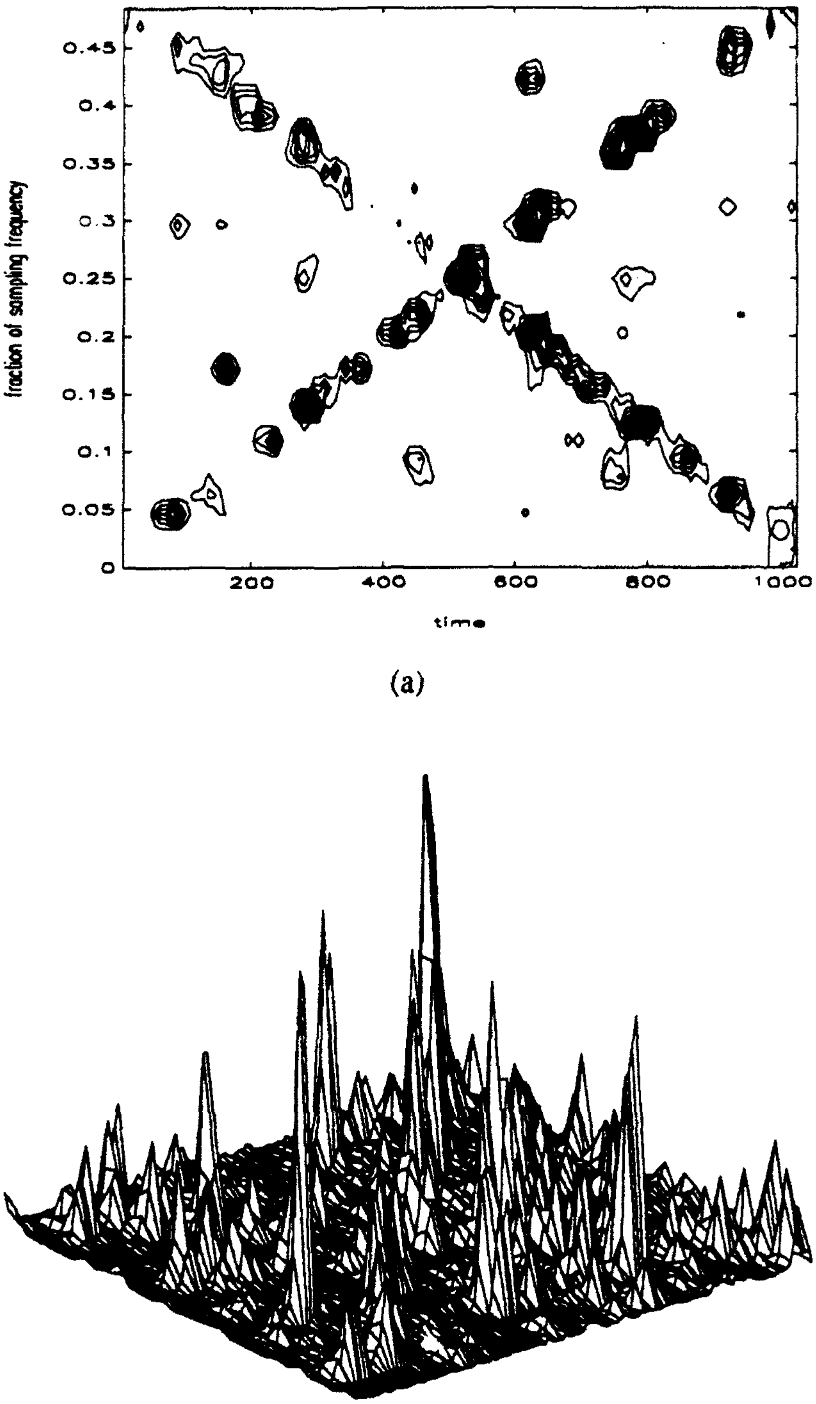
(b) Mesh plot



(a)

(b)

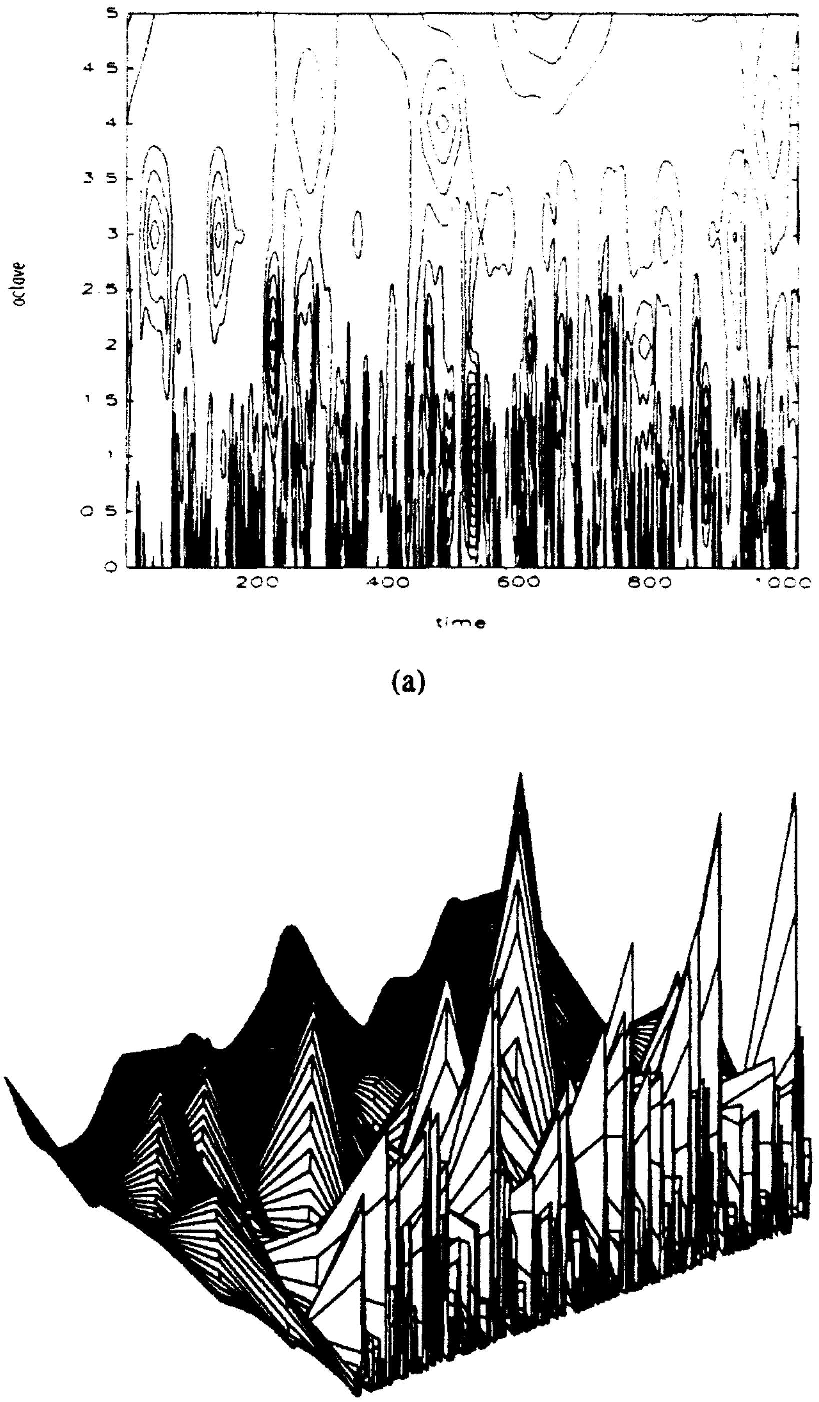
Figure 34. Wigner-Ville time-frequency distribution for two linear chirps in noise (64 point window with a 32 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

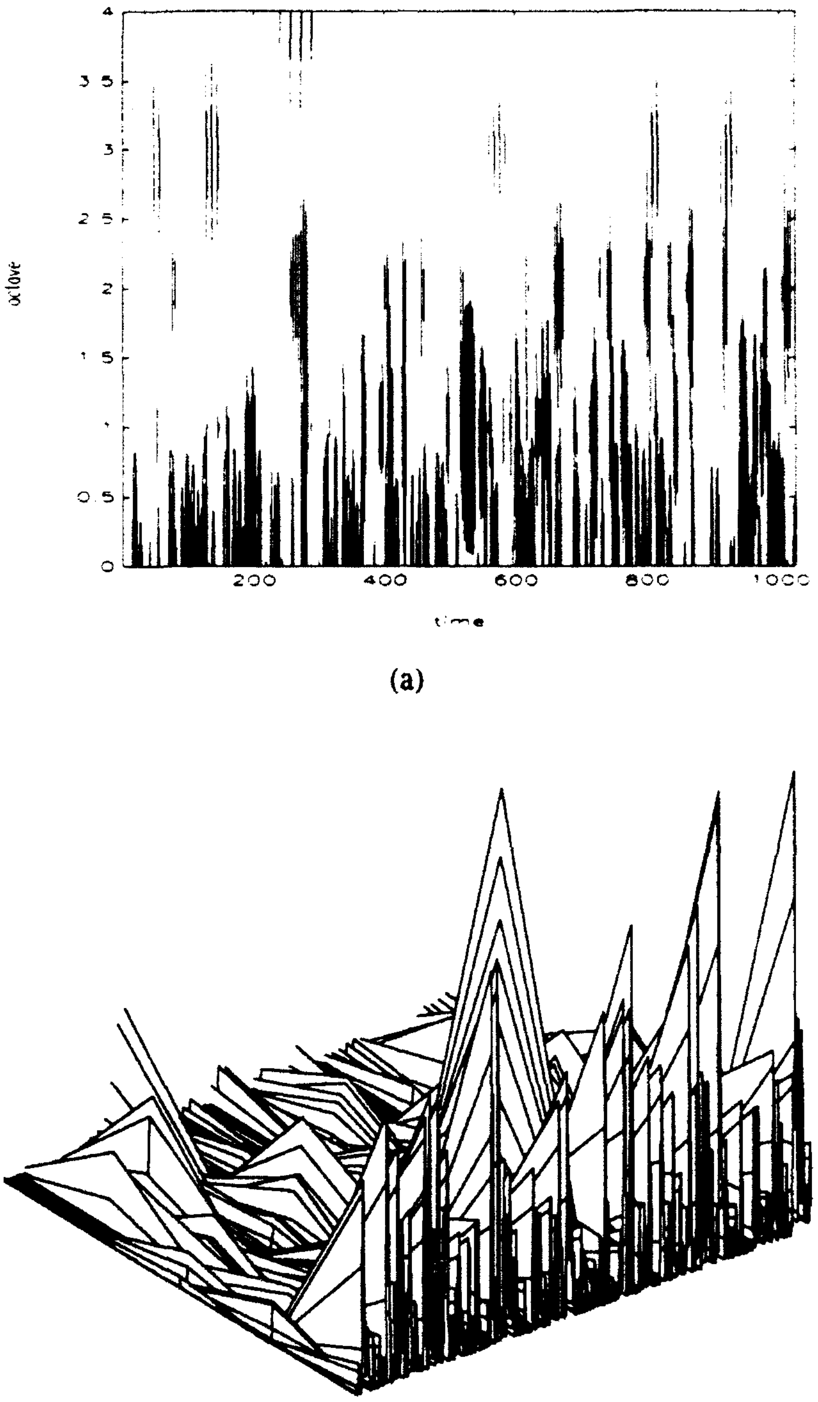
Figure 35. IPS time-frequency distribution for two linear chirps in noise (64 point window with an 8 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

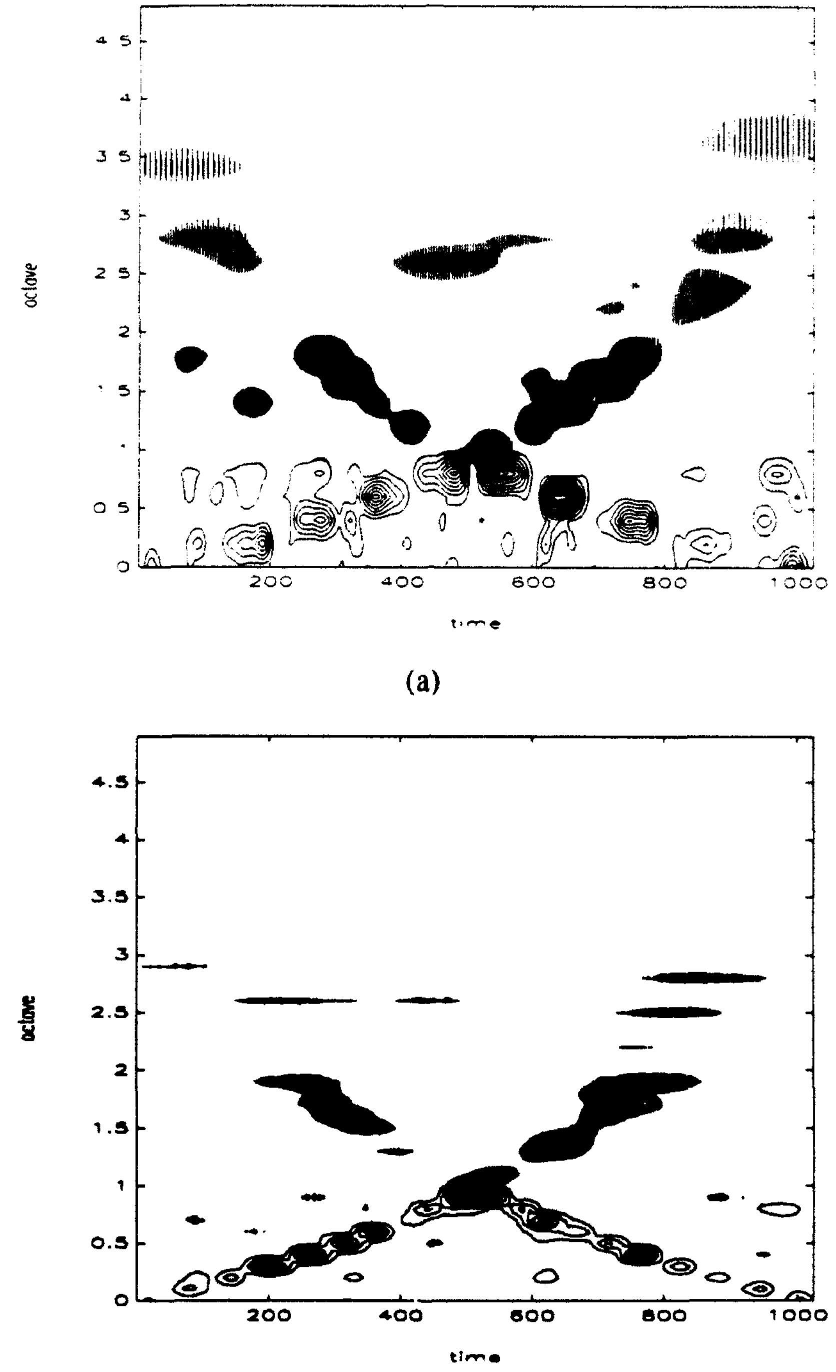
Figure 36. DCWT time-scale distribution for two linear chirps in noise
(a) Contour plot
(b) Mesh plot



(a)

(b)

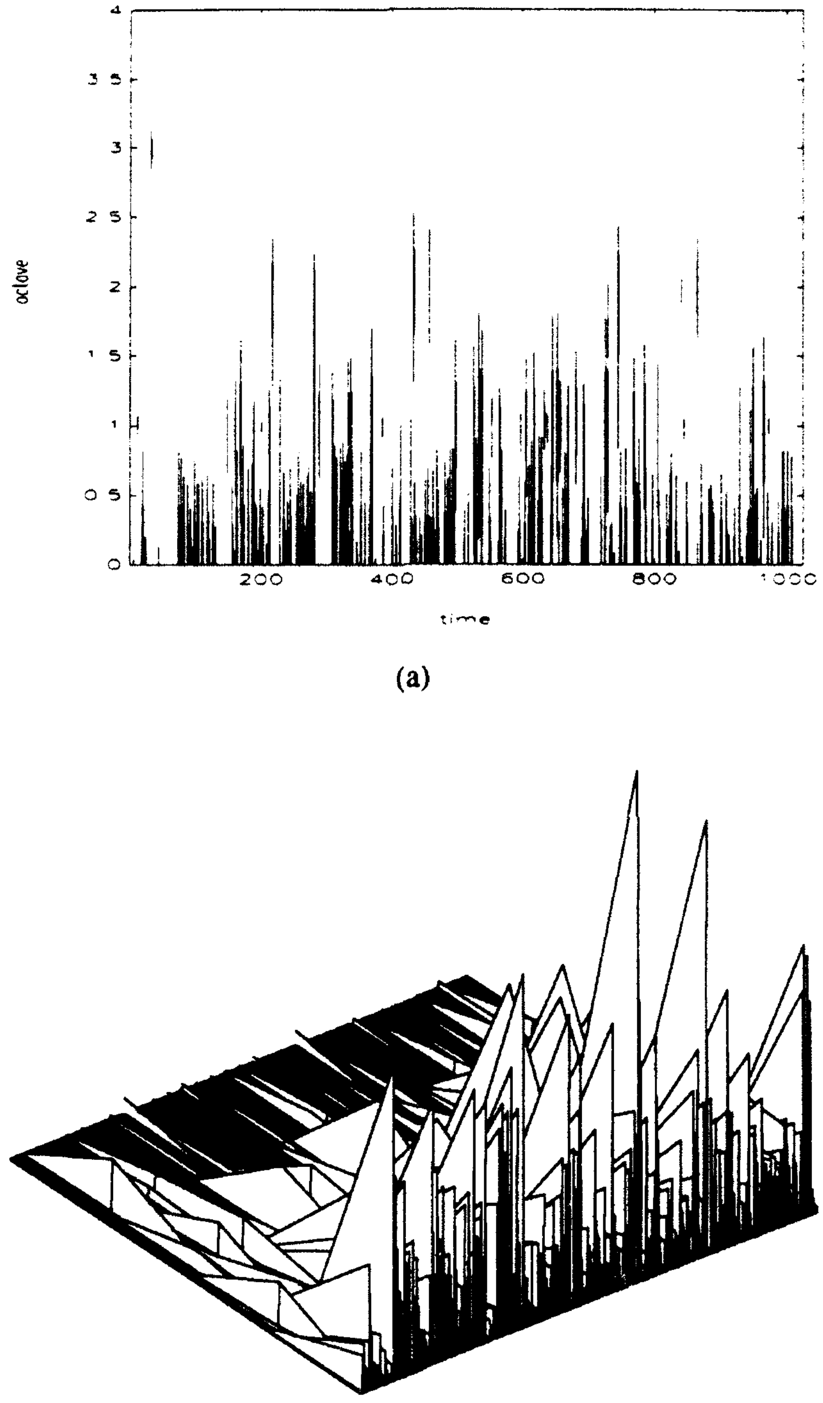
Figure 37. DWT (*à trous*) time-scale distribution for two linear chirps in noise (one voice)
(a) Contour plot
(b) Mesh plot



(a)

(b)

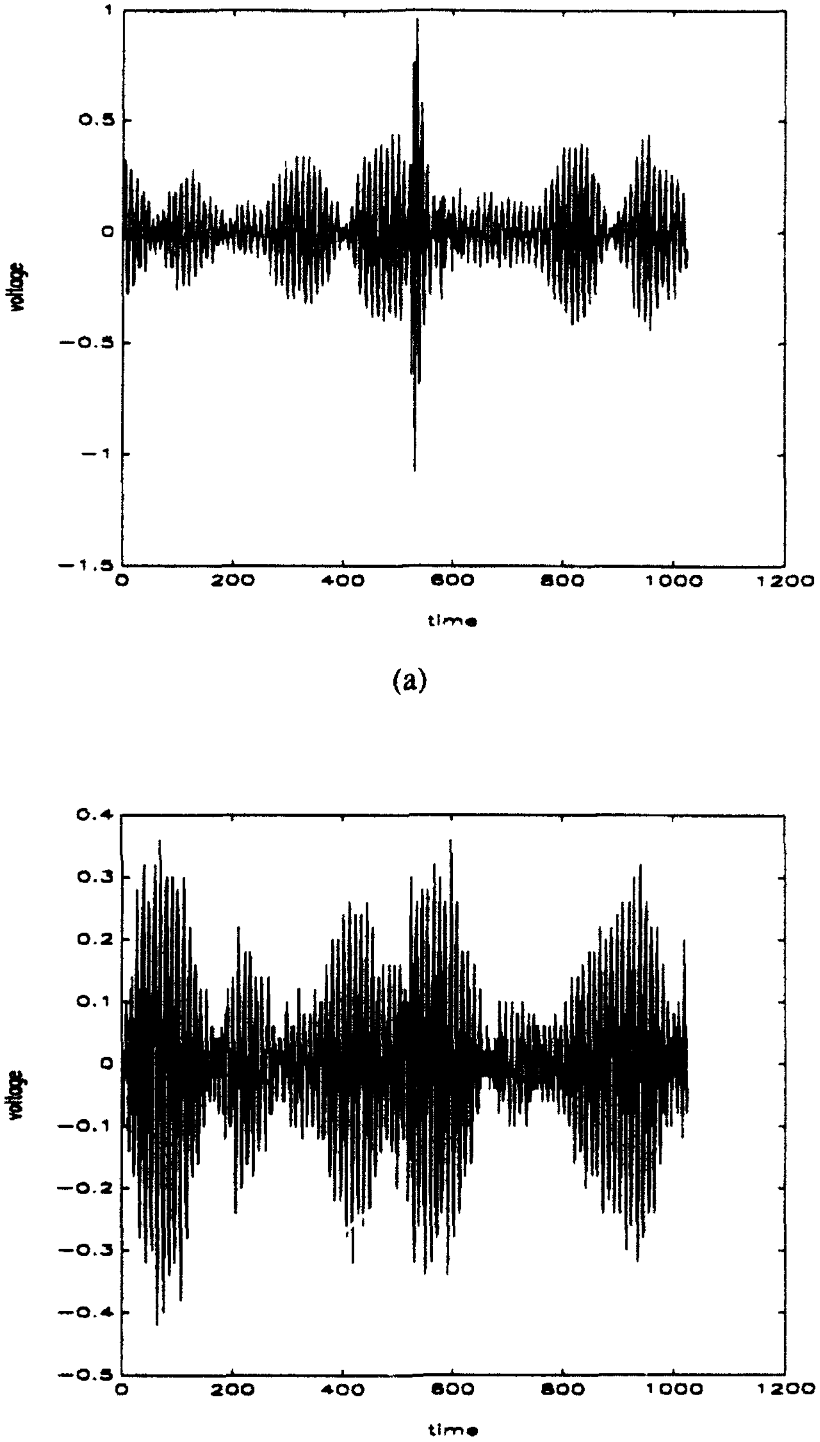
Figure 38. DWT (*à trous*) time-scale distribution for two linear chirps in noise
(a) Contour plot for 5 voices
(b) Contour plot for 10 voices



(a)

(b)

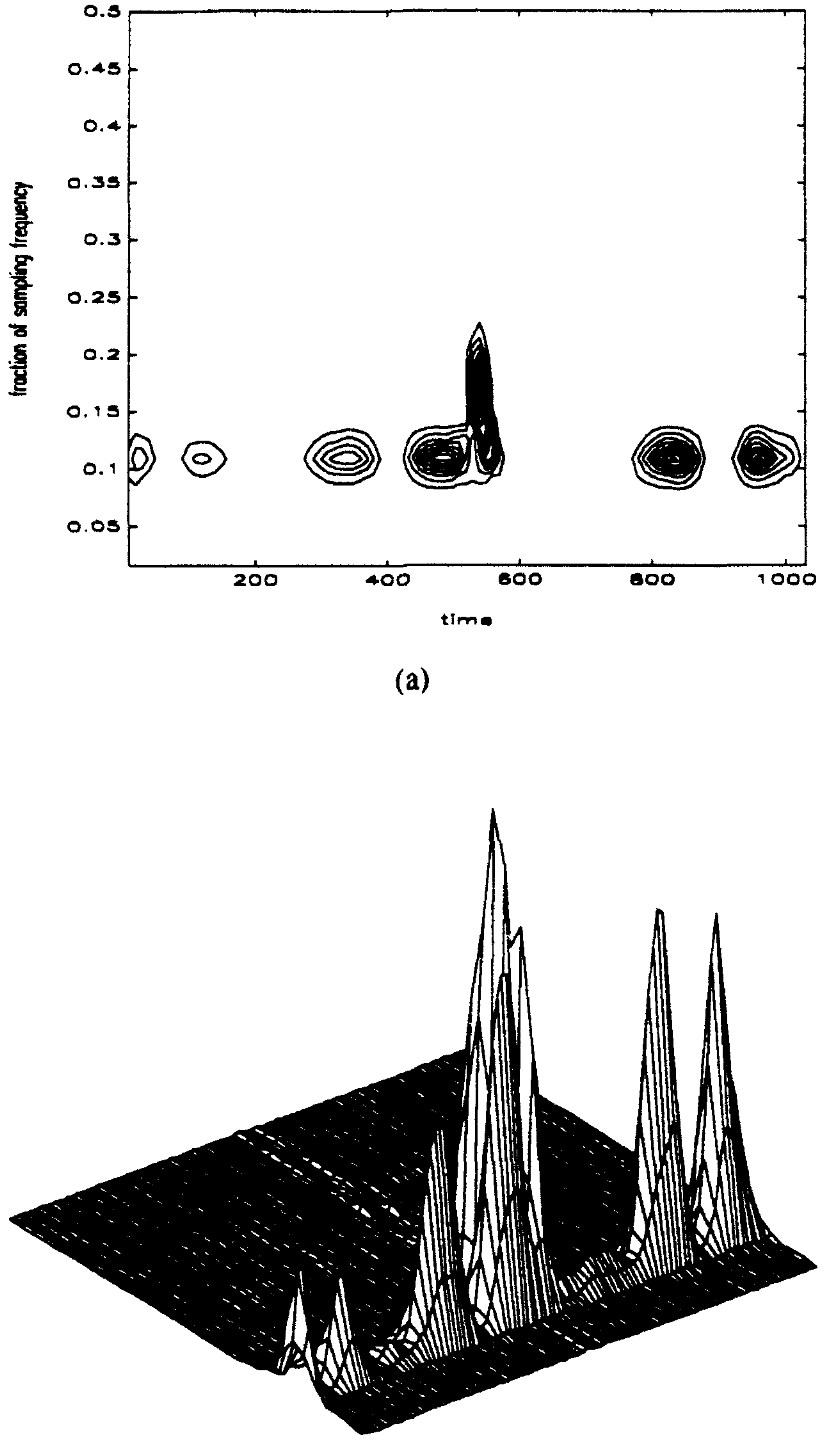
Figure 39. DWT (Mallat) time-scale distribution for two linear chirps in noise (12 point scaling function)
(a) Contour plot
(b) Mesh plot



(a)

(b)

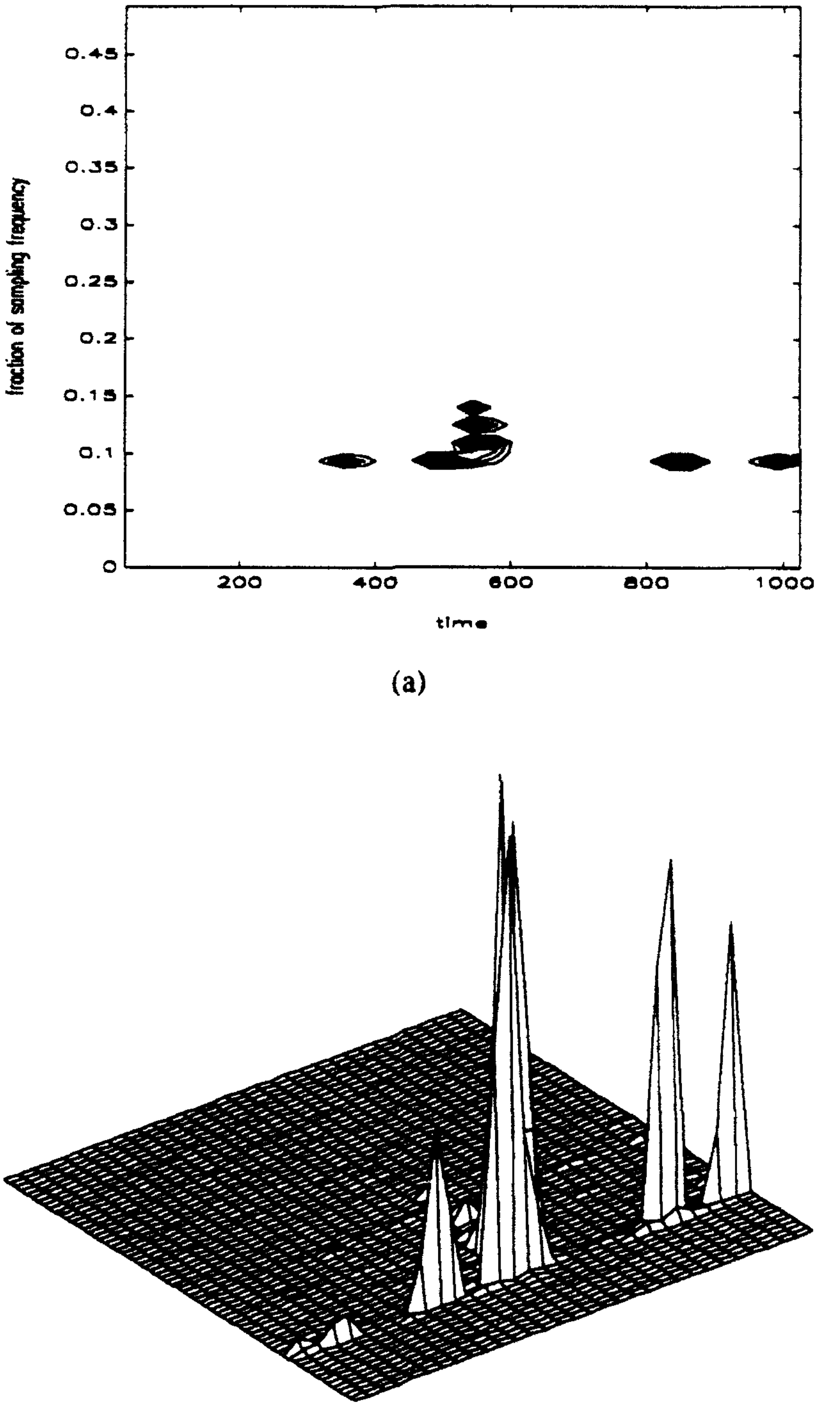
Figure 40. Raw UWB radar returns
(a) Boat with corner reflector
(b) Boat without corner reflector



(a)

(b)

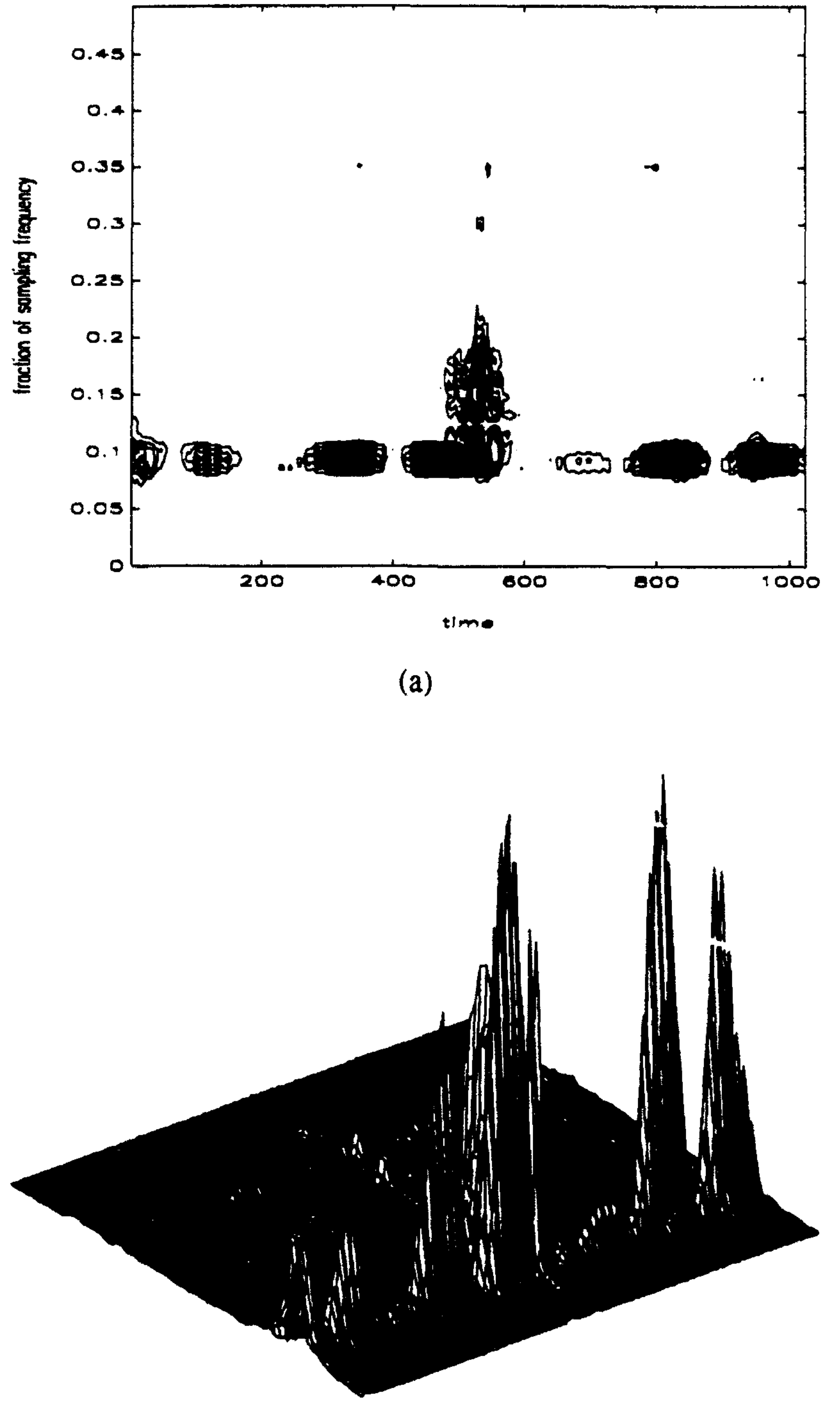
Figure 41. STFT time-frequency distribution for the boat with corner reflector (41 point Chebyshev window with a 10 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

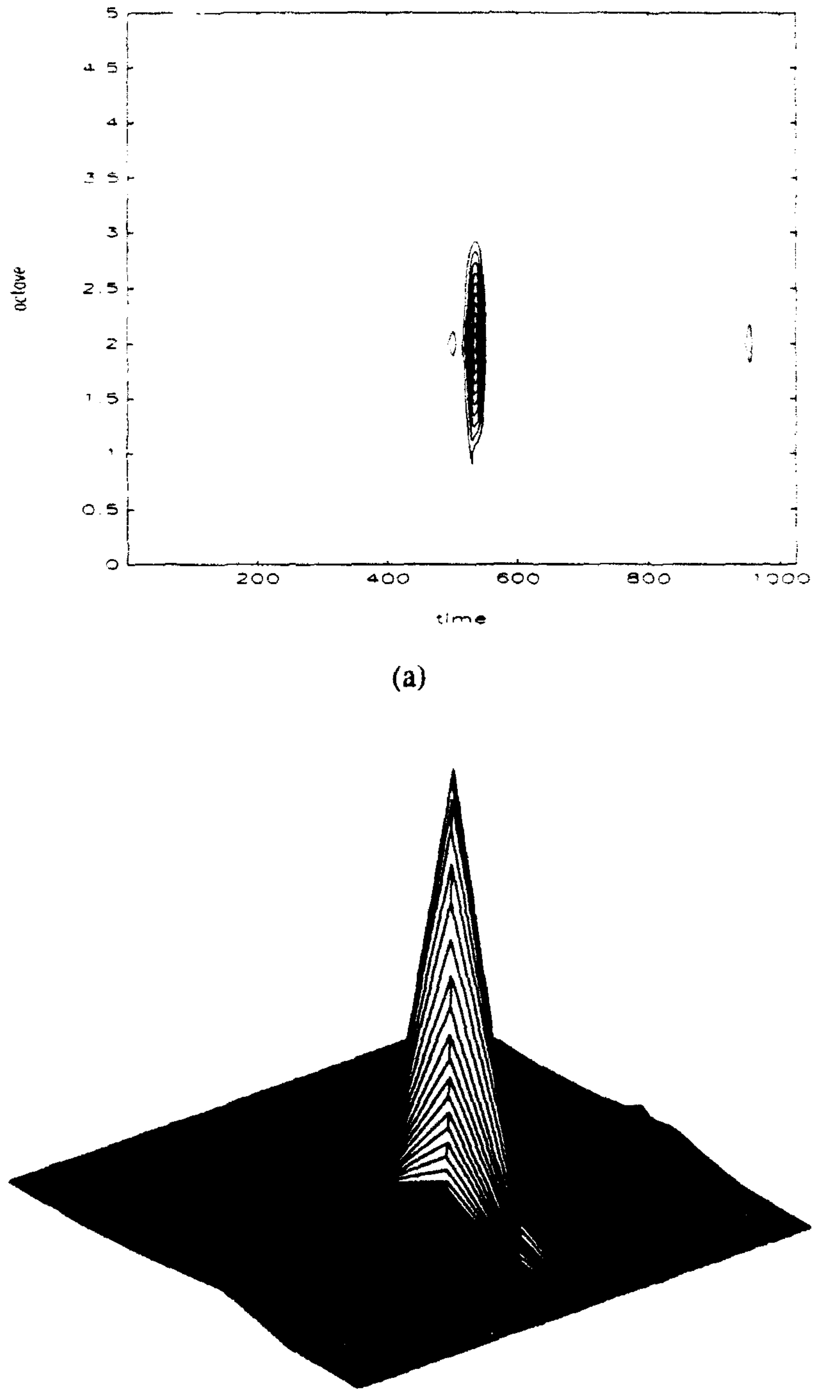
Figure 42. Wigner-Ville time-frequency distribution for the boat with corner reflector (64 point window with a 32 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

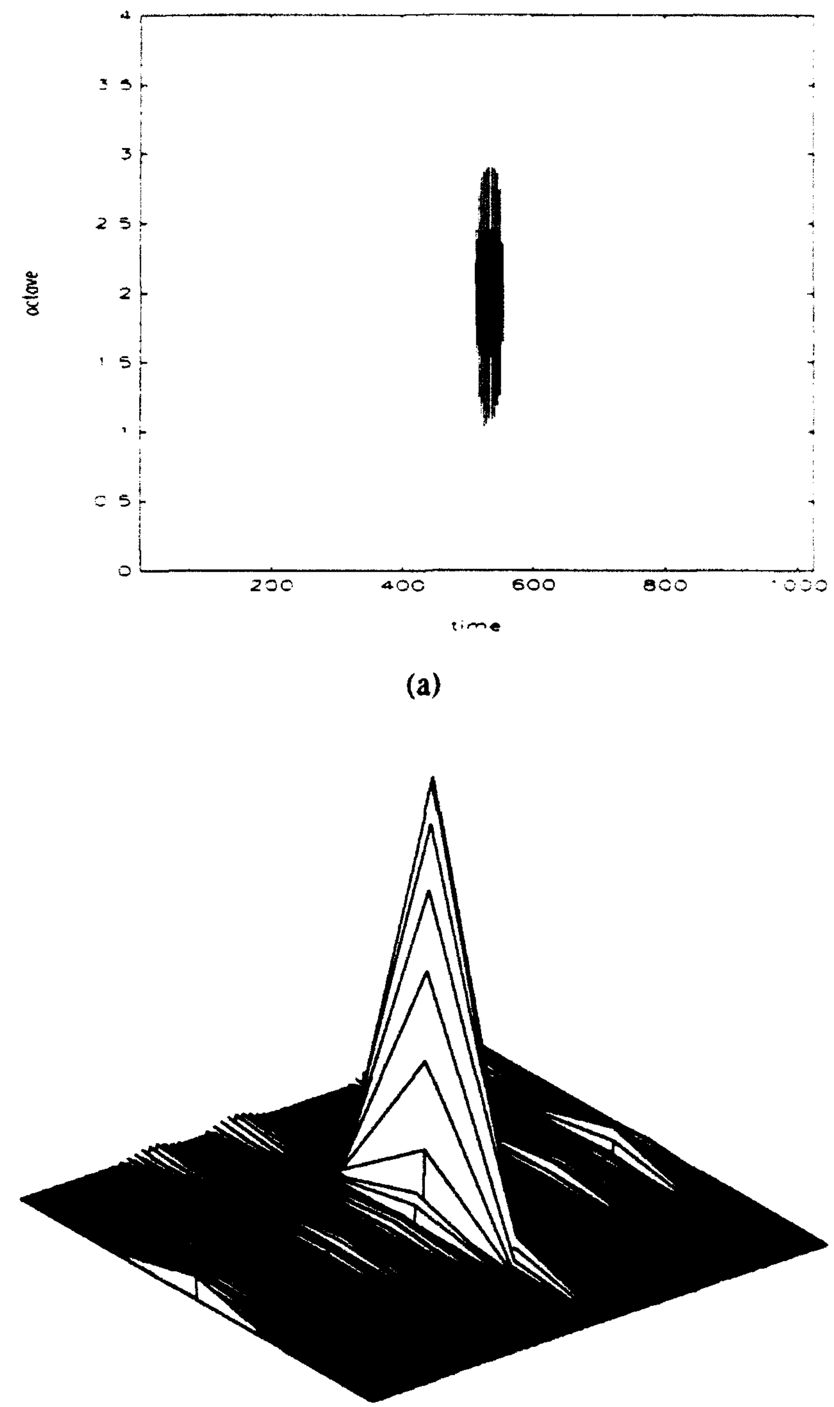
Figure 43. IPS time-frequency distribution for the boat with corner reflector (128 point window with a 4 point step)
(a) Contour plot
(b) Mesh plot



(a)

(b)

Figure 44. DCWT time-scale distribution for the boat with corner reflector (with k=π and β=0.6)
(a) Contour plot
(b) Mesh plot



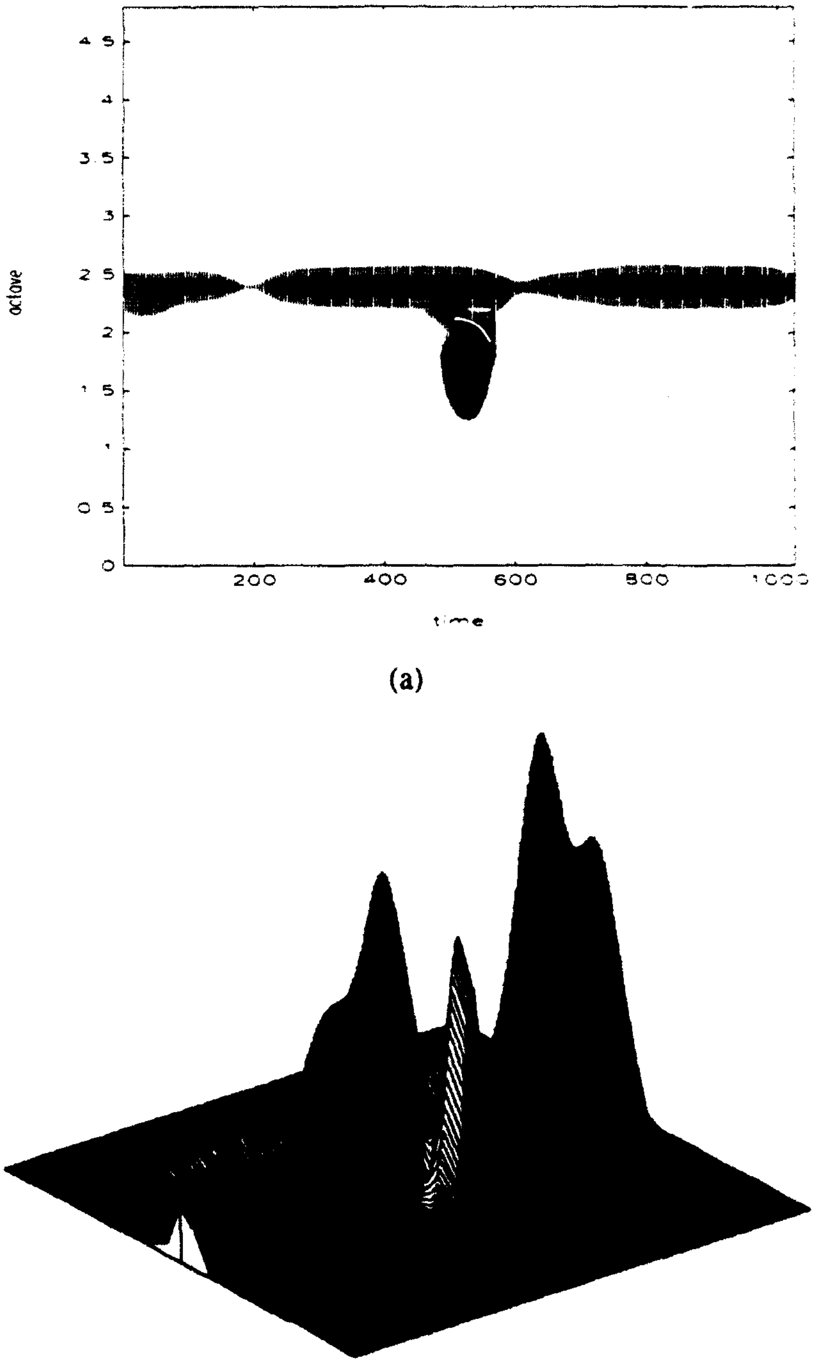
(a)

(b)

Figure 45. DWT (*à trous*) time-scale distribution for the boat with corner reflector (one voice with k=π and β=0.6)

(a) Contour plot

(b) Mesh plot



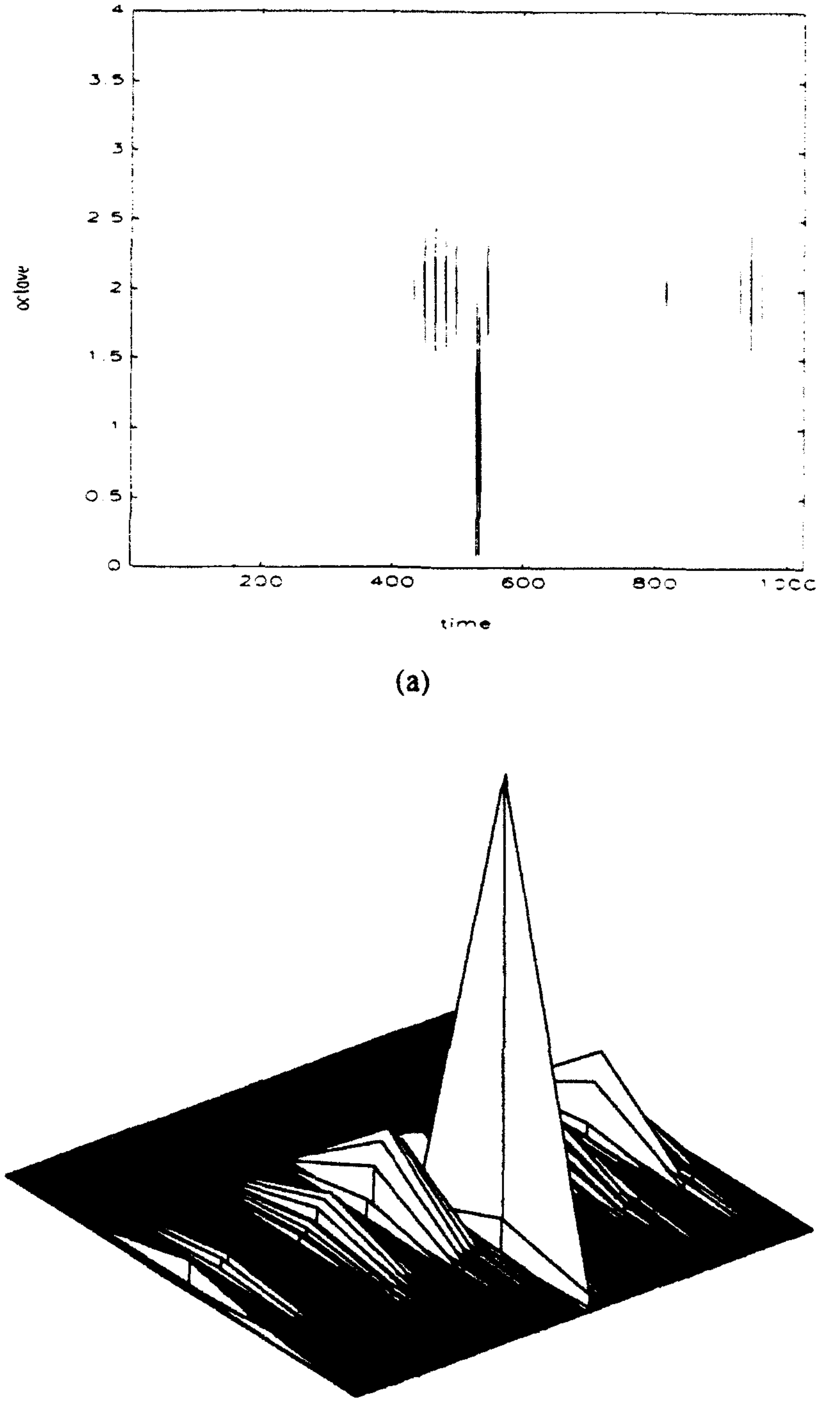
(a)

(b)

Figure 46. DWT (*à trous*) time-scale distribution for the boat with corner reflector (five voices with k=π and β=0.6)

(a) Contour plot

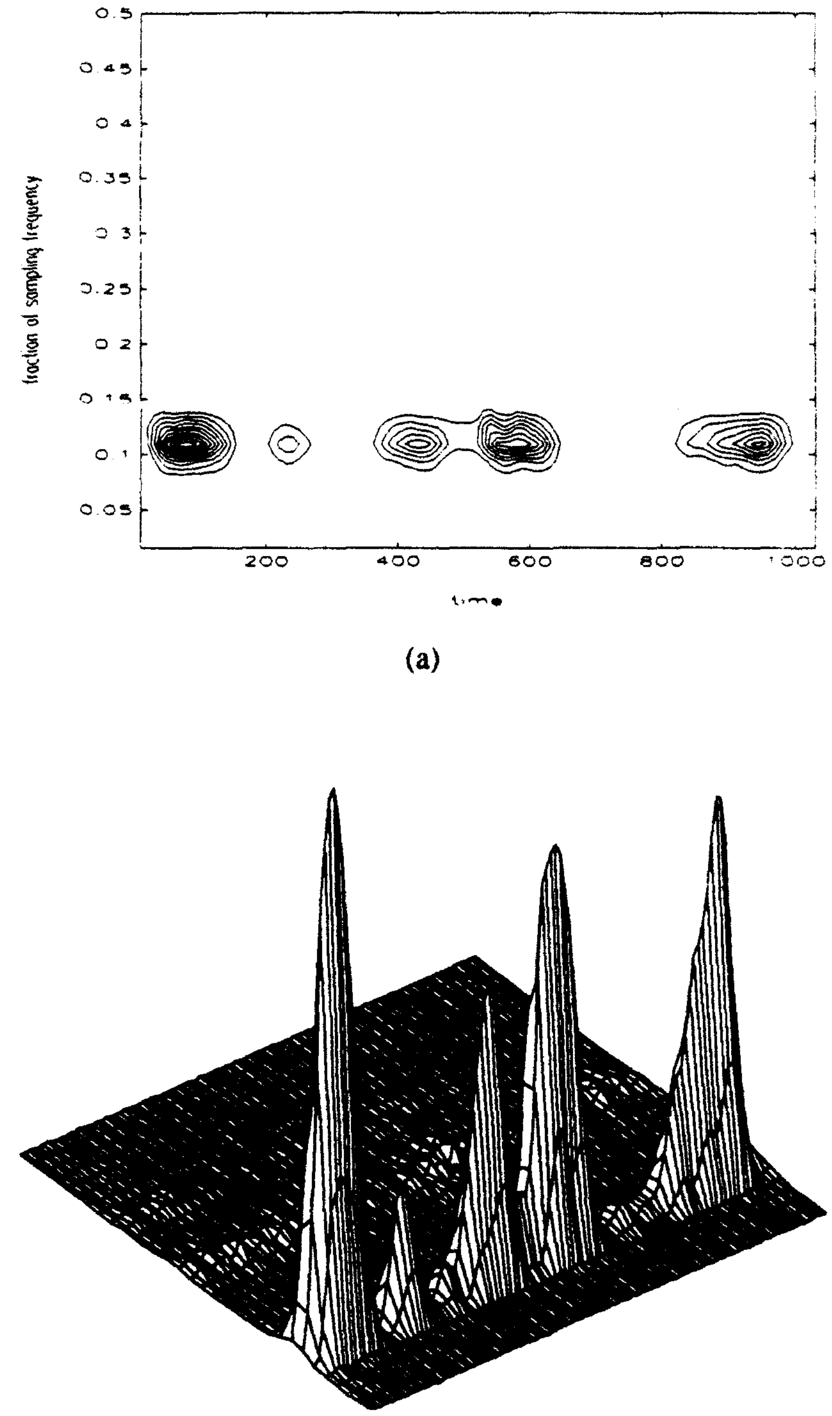
(b) Mesh plot



(a)

(b)

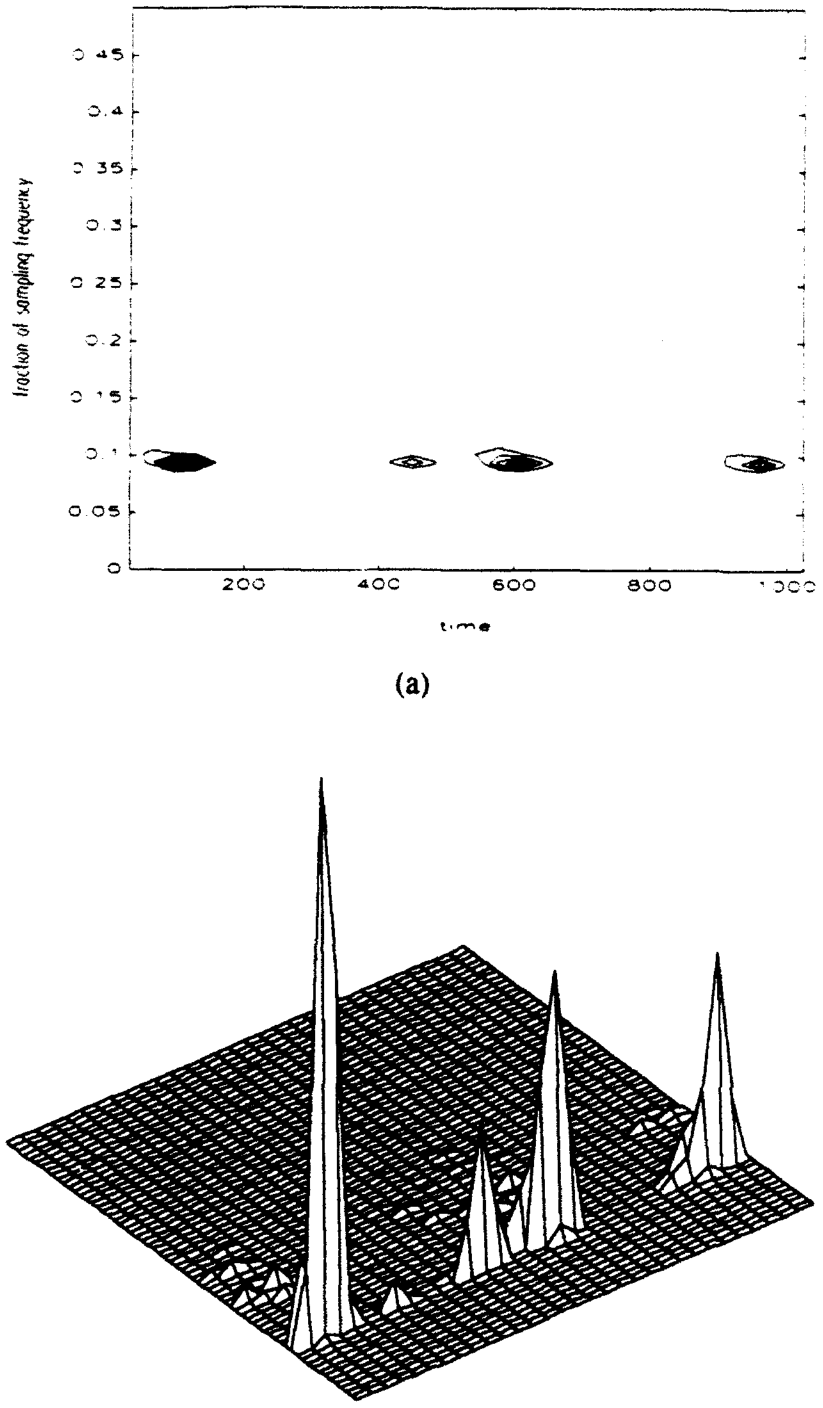
Figure 47. DWT (Mallat) time-scale distribution for the boat with corner reflector (12 point scaling function)
(a) Contour plot
(b) Mesh plot



(a)

(b)

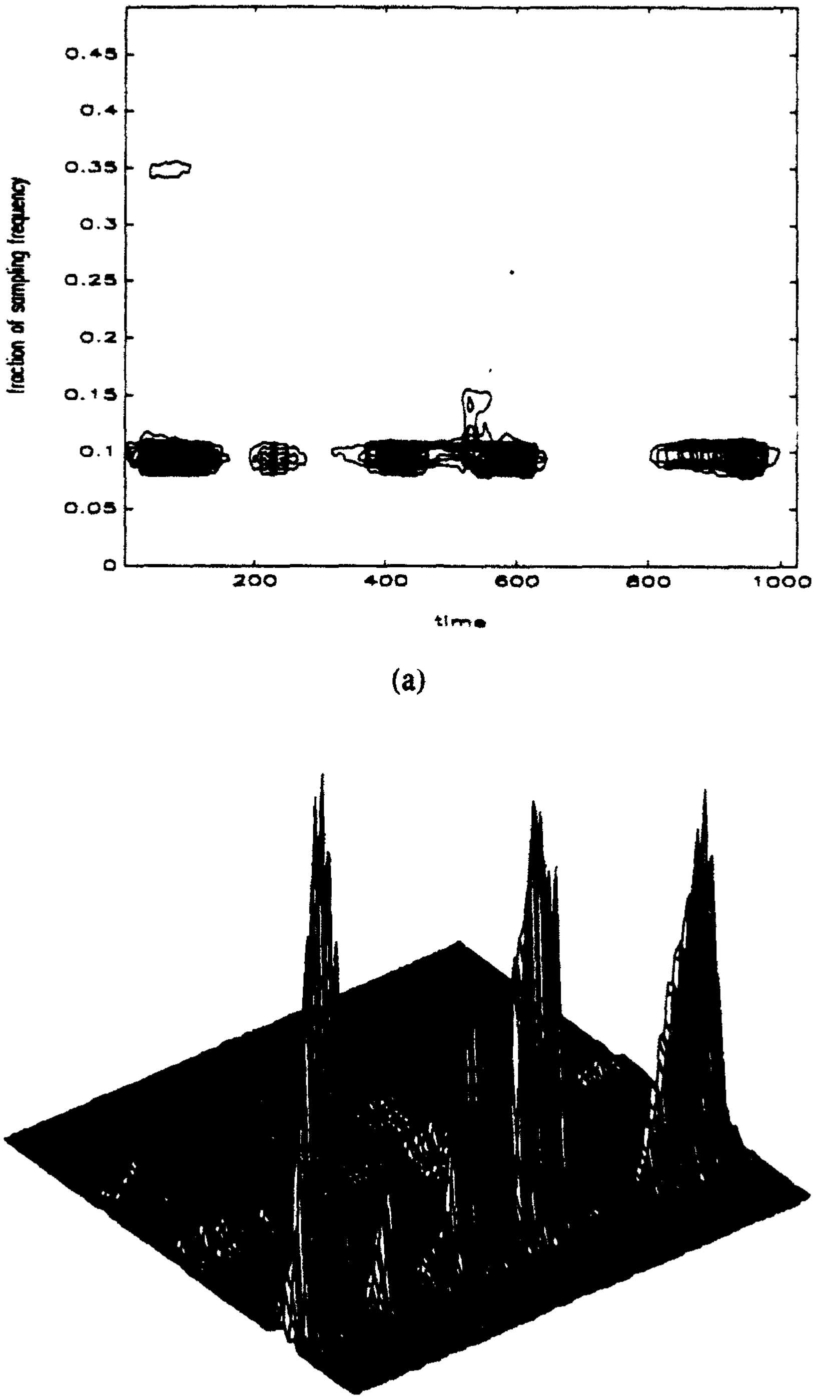
Figure 48. STFT time-frequency distribution for the boat without corner reflector
(41 point Chebyshev window with a 10 point step)
(a) Contour plot (PGR=0.0 dB)
(b) Mesh plot



(a)

(b)

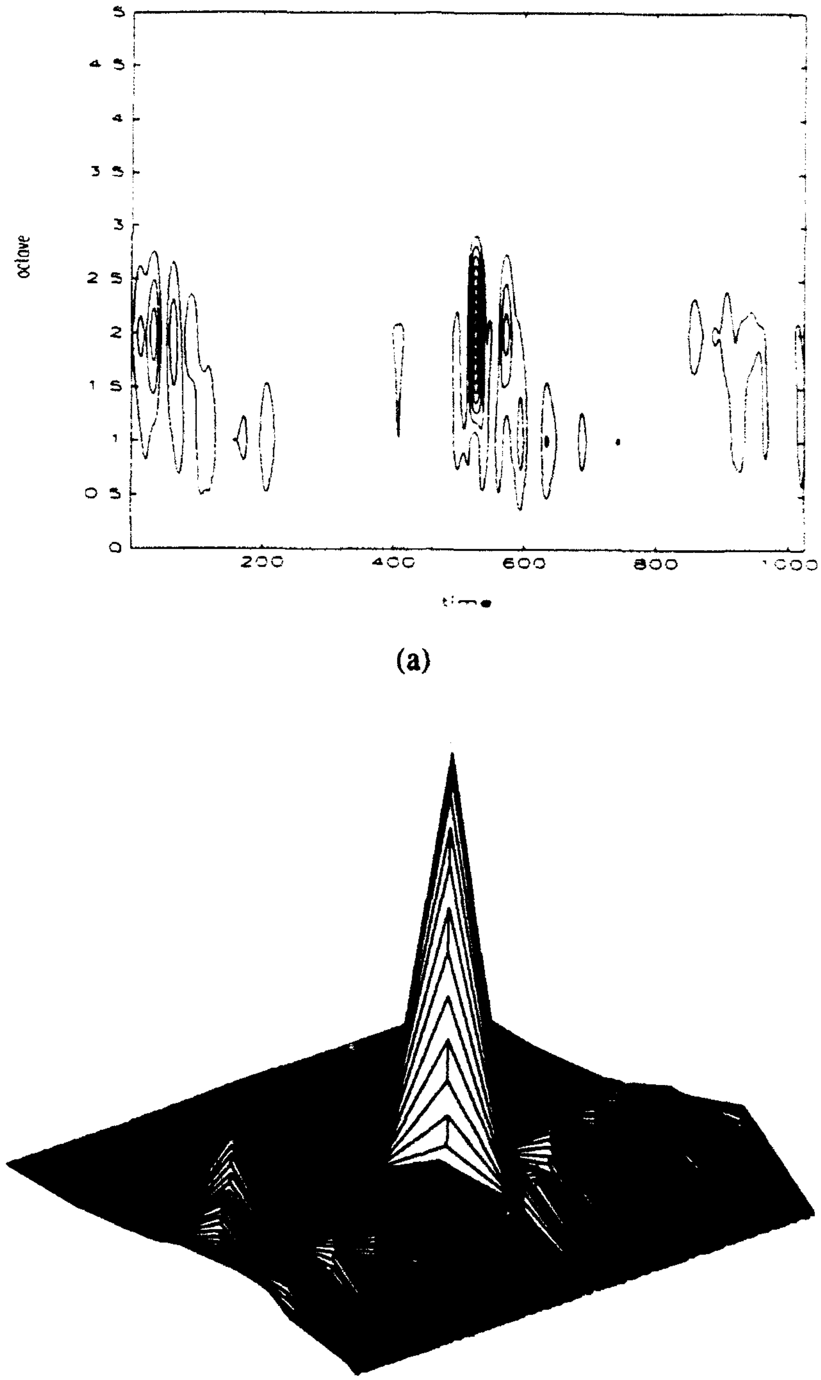
Figure 49. Wigner-Ville time-frequency distribution for the boat without corner reflector (64 point window with a 32 point step)
(a) Contour plot (PGR=0.0 dB)
(b) Mesh plot



(a)

(b)

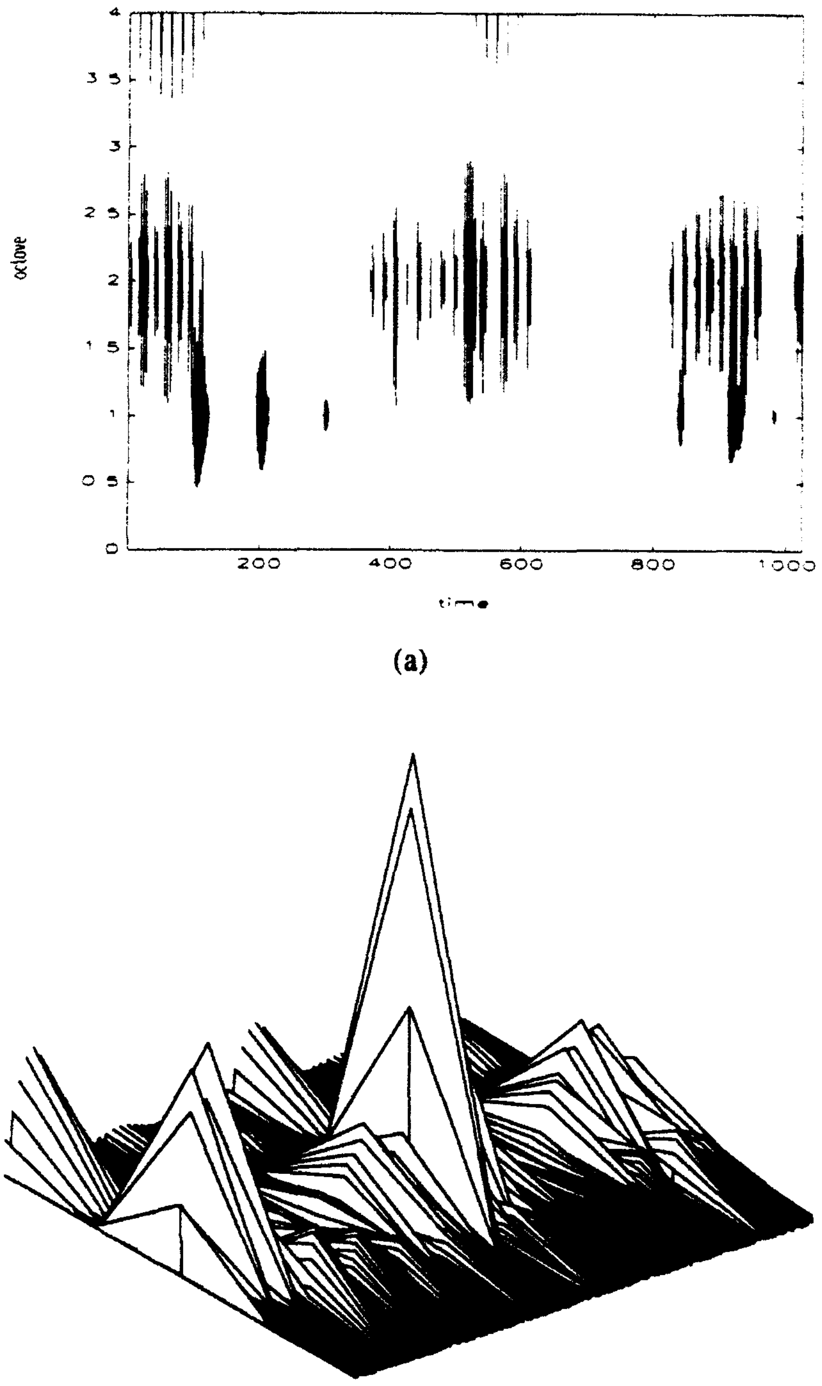
Figure 50. IPS time-frequency distribution for the boat without corner reflector
(128 point window with a 4 point step)
(a) Contour plot (PGR=0.0 dB)
(b) Mesh plot



(a)

(b)

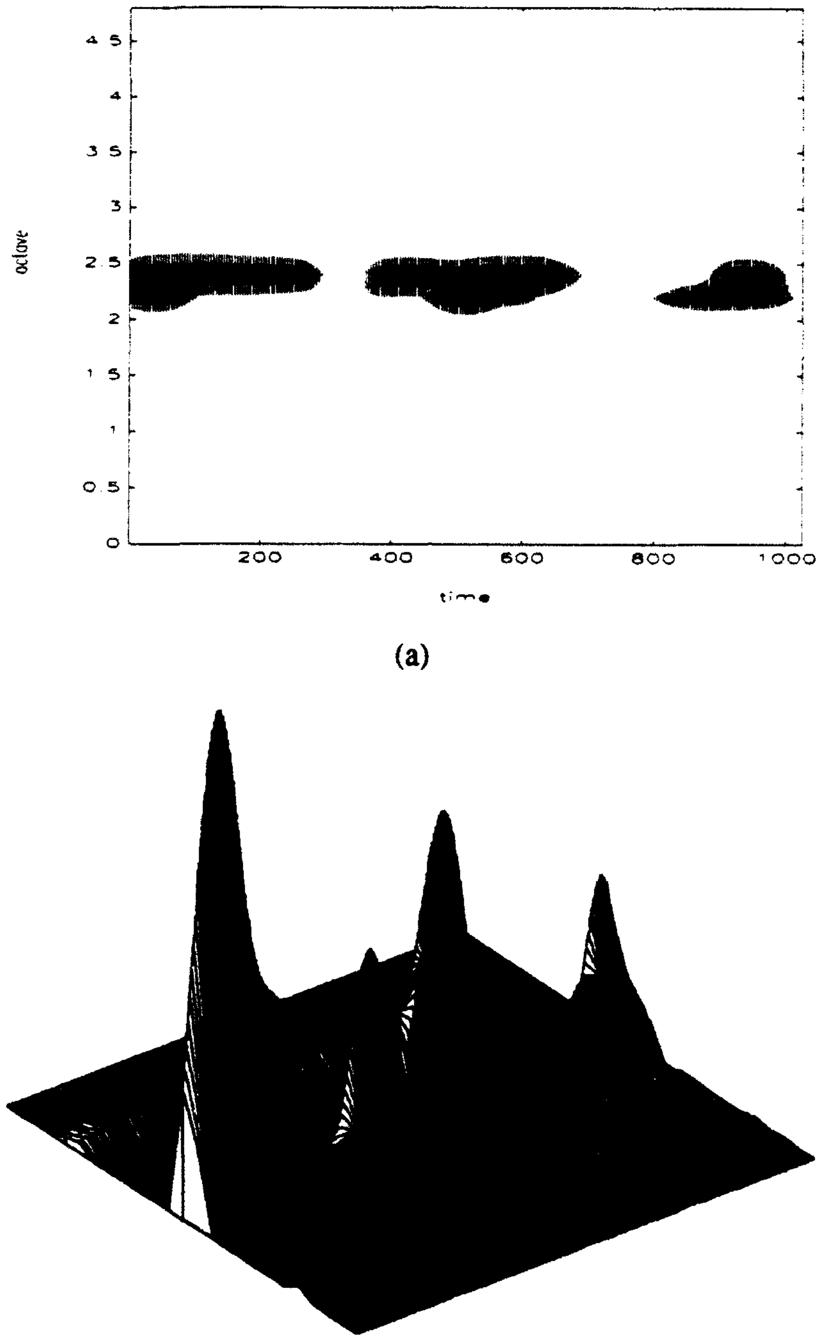
Figure 51. DCWT time-scale distribution for the boat without corner reflector (with k=π and β=0.35)
(a) Contour plot (PGR=24.59 dB)
(b) Mesh plot



(a)

(b)

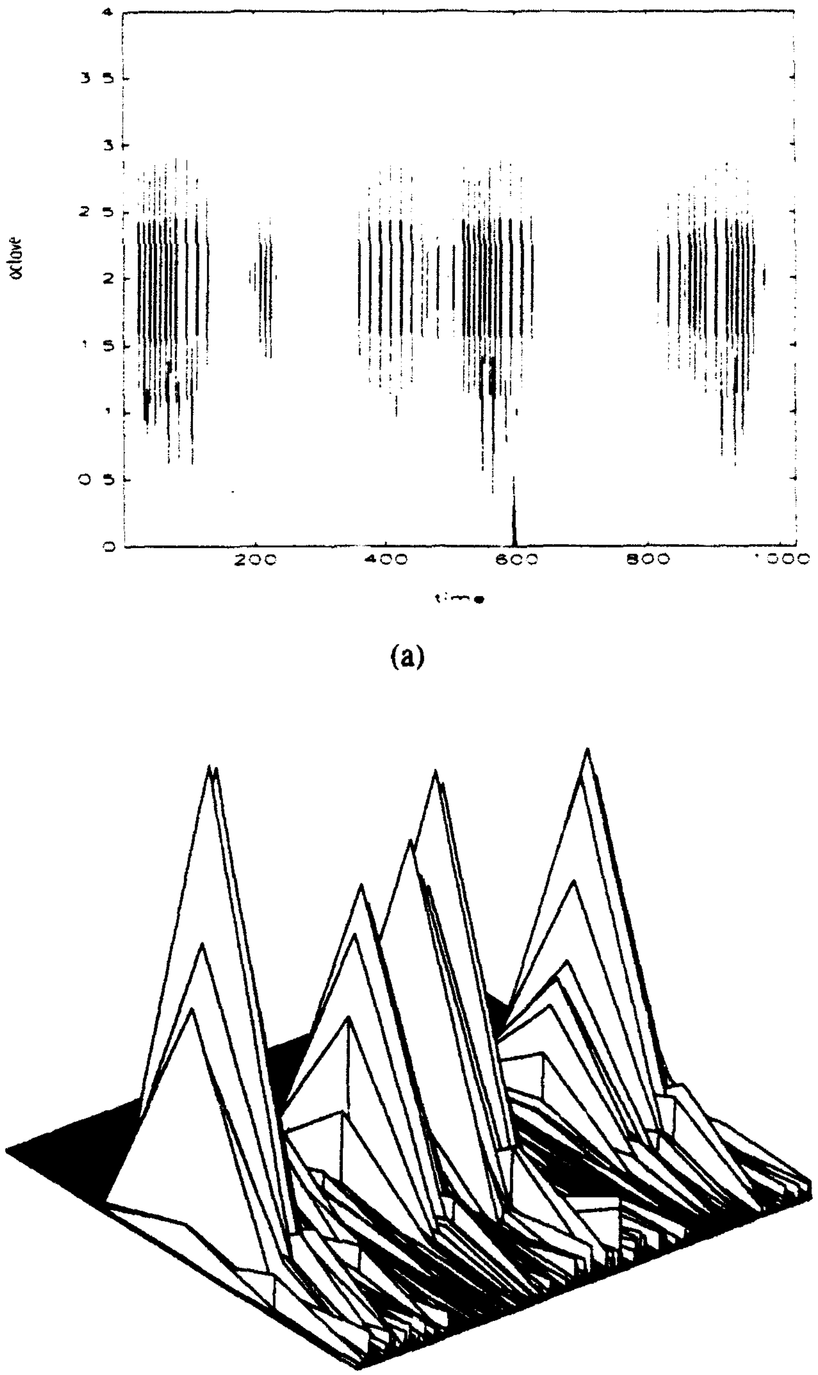
Figure 52. DWT (*à trous*) time-scale distribution for the boat without corner reflector (one voice with k=π and β=0.35)
(a) Contour plot (PGR=40.83 dB)
(b) Mesh plot



(a)

(b)

Figure 53. DWT (*à trous*) time-scale distribution for the boat without corner reflector (five voices with k=π and β=0.35)
(a) Contour plot (PGR=0.0 dB)
(b) Mesh plot



(a)

(b)

Figure 54. DWT (Mallat) time-scale distribution for the boat without corner reflector (12 point scaling function)
(a) Contour plot (PGR=0.0 dB)
(b) Mesh plot

VI. RECOMMENDATIONS AND CONCLUSIONS

In this thesis, the performance of time-frequency and time-scale techniques on synthetic test signals and raw ultra-wideband radar (UWB) data has been examined. Traditional time-frequency methods such as the Short Time Fourier Transform (STFT), the Wigner-Ville distribution (WD), the Instantaneous Power Spectrum (IPS), and their applications to non-stationary signal analysis were presented in Chapter III. Time-scale methods such as the "Discrete" Continuous Wavelet transform (DCWT), *à trous* discrete wavelet transform (DWT) algorithm and, Mallat's DWT algorithm were derived in Chapter IV.

The STFT is a spectral analysis technique that relies on the Fourier transform to compute the spectrum of a windowed data segment. The method consists of multiplying a signal *s(t)* with a compactly supported window *g(t)* centered around an arbitrary point. The magnitude squared Fourier coefficients of the windowed coefficients provide an indication of the frequency content of a signal in the vicinity of the arbitrary point. The power spectral density (PSD) of the signal is obtained by sliding the window across the data record at regular finite intervals, and computing the magnitude squared of the Fourier coefficients. This process uses a single analysis window that fixes time-frequency resolution across the time-frequency surface and is well suited for analyzing signals consisting of a few stationary components with spectral descriptions that evolve slowly with time.

The Wigner-Ville distribution and IPS are based on the Wiener-Khinchine theorem which relates the PSD to the autocorrelation function of a band limited, wide sense stationary random process. The WD uses a time-dependent autocorrelation function and IPS uses an averaged autocorrelation function. Both methods provide valid estimates of the PSD but also have a fixed time-frequency resolution across the time-frequency distribution. As with the STFT, this occurs because the width of the finite analysis window does not vary with frequency (i.e., the window maintains a constant absolute bandwidth in the frequency domain). Therefore, the STFT, WD and IPS methods are limited in non-stationary applications because abrupt changes in signal behavior cannot be simultaneously analyzed with long duration windows required for good frequency resolution, and short duration windows required for good temporal resolution.

For the spectral analysis non-stationary signals, wavelet transforms provide an attractive alternative to conventional time-frequency methods. This is true because the absolute bandwidth of the analysis window (i.e., the analyzing wavelet) varies with frequency. At high frequencies the window shortens, allowing good time resolution, and at low frequencies the window broadens, offering good frequency resolution. To satisfy the uncertainty principle, the concept of a variable absolute bandwidth in wavelet transforms leads to the notion of a fixed relative bandwidth in the analyzing wavelet. This implies that the time-bandwidth product across the time-frequency surface remains constant, however the wavelet transform provides good time resolution (i.e., poor frequency resolution) at high frequencies and good frequency resolution (i.e., poor time resolution) at low frequencies [13].

Each DWT technique discussed in Chapter IV has different properties. The DCWT is an undecimated, non-orthogonal DWT. The *à trous* discrete wavelet transform is a non-orthogonal decimated transform, and Mallat's algorithm is an orthogonal, decimated version of the DWT. Non-orthogonal wavelets are desirable because the bandwidth of the analyzing wavelet may be chosen by the user. However, non-orthogonal DWTs are not invertible because the DWT coefficients computed at each octave are not independent and contain redundant information from octave to octave. In some applications, orthogonal wavelets are desirable because the original signal may be reconstructed from the DWT coefficients. However, the major drawback of orthogonal wavelets is a lack of a flexible filter design which results in a relative bandwidth fixed at $\pi/2$.

The *à trous* and Mallat's DWT algorithms are multiresolution algorithms that may be implemented with similar filter bank structures. The feature that distinguishes the *à trous* algorithm from Mallat's algorithm is the choice of filters; low pass filters f and h (also referred to as the scaling functions) and, the high pass filter g (also known as the analyzing wavelet). For orthogonal wavelet transforms, the analyzing wavelet g is determined directly from the scaling function h. For non-orthogonal wavelet transforms, g is independent of the low pass filter f and, must satisfy the analyzing wavelet properties listed in Chapter IV.

The multiresolution filter bank implementation of the *à trous* and Mallat's algorithms are shown in the two step algorithms shown in Figure 4. First, the DWT coefficients are computed at octave i [2] by convolving the signal s with analyzing wavelet g. Second, the signal is scaled for the next octave by convolving s with the scaling function (f for the non-orthogonal version and h for the orthogonal version). This procedure repeats itself as the scaled version of s is again filtered by the high pass filter, and scaled by the low pass filter through the range of desired octaves.

A comparison of the performance of the time-frequency and time-scale algorithms for five test signals and two raw UWB target returns embedded in non-stationary background interference is provided in Chapter V. The radar returns consist of a small boat with and without a corner reflector, and each data record consists of 1024 points. The time-frequency/scale techniques show good time and frequency resolution for the dirac and complex sinusoid. The scalograms for the DCWT, one voice *à trous* and Mallat's algorithm for the single linear chirp show inferior performance when compared to the time-frequency methods, however, the multiple voice *à trous* method shows comparable resolutions at high scales (octaves). The STFT, IPS and multi-voice *à trous* techniques show superior performance for the synthetic signal consisting of two crossing linear chirps. However, only IPS and the multiple voice *à trous* algorithm are able to discern the two crossing chirps embedded in 0 dB white Gaussian noise.

The results are described in terms of a processing gain ratio (PGR). The PGR is computed in decibels (dB) and is defined as the voltage ratio of the maximum voltage value (V_{max}) in the time-frequency/time-scale surface divided by the mean voltage value (V_{mean}) of the surface. For each UWB radar record, the location of the target was at approximately time bin 520. Because of the existence of high levels of interfering noise at time bin greater than 100 and time bin less than 1000, the PGR was arbitrarily set equal to zero if V_{max}occurred in those bins. To resolve this problem, future study should include developing a detector and detection criteria capable of distinguishing a target return in time bins 1-1024.

The STFT, WD, IPS and the wavelet transform techniques demonstrate sufficient joint time-frequency resolution to unambiguously differentiate the raw UWB radar signal corresponding to the boat with a corner reflector. However, the results indicate that only IPS, the DCWT and the one voice *à trous* algorithms may be used to successfully detect the transient for the data corresponding to the boat without th^ corner reflector. However, IPS achieves a very low processing gain when compared to the wavelet transform methods. The wavelet transform performs well for the boat without corner reflector data because, at high frequencies, the analyzing wavelet contracts to level where it provides a better match to the transient nature of the target return.

APPENDIX A. MATLAB SOURCE CODE

The algorithms described in Chapter III and IV were implemeted using MATLAB software. The routines that gernerated the distributions examined in Chapter V are listed below.

1. Short Time Fourier Transform

```
function [spdatamatrix]=stft(data,spacing,p);
% Filename:  stft.m
% Title: Spectrogram using the Short Time Fourier Transform (STFT)
% Date of Last Revision: 04 Dec 92
% Comments: This program computes the STFT with a 41 point
%                Chebyshev window with a 10 point sliding window.
%
% Input Variables:
%     data:      input data
%     spacing:  overlap (10)
%      p:          p=1 indicates contour plot
%                   p=2 indicates mesh plot
%
% Output Variables:
%     datamatrix:  The STFT time-frequency surface
%
% Associated MATLAB Files: pgr.m
%
% Associated Functions: none
clg
format compact
xt=data(1,:);
a=chebwin(41,30);
recwin=zeros(1,41)';
recwin(21:41)=data(1:21);
window=a.*recwin;
datamatrix(:,1)=window;
recwin=zeros(1,41)';
recwin(11:41)=data(1:31);
window=a.*recwin;
datamatrix(:,2)=window;
endpoint=length(data);
stop=fix(endpoint/spacing)-2;
for index=2:stop;
  recwin=data((index*spacing)-19:(index*spacing)+21);
  window=a.*recwin;
  datamatrix(:,(index+1))=window;
end
recwin=zeros(1,41)';
recwin(1:31)=data(endpoint-30:endpoint);
```

```
window=a.*recwin;
datamatrix(:,stop+2)=window;
recwin=zeros(1,41)';
recwin(1:21)=data(endpoint-20:endpoint);
window=a.*recwin;
datamatrix(:,stop+3)=window;
size(datamatrix);
spdatamatrix1=fft(datamatrix,64);
spdatamatrix=spdatamatrix1(1:32,:);
size(spdatamatrix)
%Calculate PGR
[MMM,NNN]=size(spdatamatrix1);
spdatamatrix(1:MMM,1:NNN/2)=spdatamatrix1(1:MMM,1:NNN/2);end;
[PGR,ii,jjjj,maxmwt]=pgr(spdatamatrix);
[MM,NN]=size(spdatamatrix);
if p==1,
 contour(flipud(abs(spdatamatrix).^2),10,(1:NN)*spacing,(1:MM)/2/MM)
 title(['PGR=',num2str(PGR),' dB'])
 xlabel('time')
 ylabel('fraction of sampling frequency')
end
if p==2,
 mesh(flipud(abs(spdatamatrix).^2))
end
```

2. Wigner-Ville Distribution

```
function P = wvd(data,winlen,step,begin,theend,p)
% Filename:  wvd.m
% Title: The Wigner-Ville Dstribution
% Date of Last Revision: 04 Dec 92
% Comments: This program computes the Wigner-Ville Distribution
%               for the input data sequence. Window length and time step size
%               are determined by the user but the window length should be a
%               power of two. By default the entire data sequence is used but
%               user may specify specific intervals within the data by using
%               'begin' and 'end'.
%
% Input Variables:
%     data:       input data
%     spacing:  overlap (10)
%     data:       input data sequence
%     winlen:   window length
%     step:        time step size
%     begin:      desired starting point within data
%     theend:    desired ending point within data
%     p:             p=1 indicates contour plot
%                    p=2 indicates mesh plot
%
% Output Variables:
%     P:             The Wigner-Ville time-frequency surface
%
```

```
% Associated MATLAB Files: pgr.m
%
% Associated Functions: none

clg
[m.n] = size(data);
if m > n
  data = data.';
end
datalen = length(data);
% use user specified starting and ending points if present
start = 1;
finish = datalen;
if nargin == 5
  if begin > 1
    start = begin;
  end
  if theend < datalen
    finish = theend;
  end
end
% interpolate data
% data = interp1(data,2*datalen);
%  datalen = datalen*2;
% initialize data spaces
data = [zeros(1,winlen/2) data zeros(1,winlen/2)];
prod = zeros(1,winlen/2 + 1);
corr = zeros(1,winlen);
PS = zeros(1,winlen);
index = 1;
for n = (winlen/2)+start:step:(winlen/2)+finish
  prod = data(n-winlen/2:n).*conj(fliplr(data(n:n+winlen/2)));
  corr = [prod conj(fliplr(prod(2:winlen/2)))];
  corr(1) = 0;
  PS(index,:) = fft(corr);
  index = index + 1;
end
PS=fliplr(PS);
[PGR,ii,jjjj,maxmwt]=pgr(PS);
[mmm,nnn]=size(PS)
if p==1,
contour(abs(PS').^2,20,(1:mmm)*step,(0:nnn-1)/2/winlen);
xlabel('time')
ylabel('fraction of sampling frequency')
end
if p==2,
mesh(abs(PS').^2)
end
```

3. Instantaneous Power Spectrum

```
function P =ips(data,wintype,winlen,step,pp)
```

```
% Filename: ips.m
% Title: The Instantaneous Power Spectrum
% Date of Last Revision: 04 Dec 92
% Comments: This function computes an Instantaneous Power Spectral (IPS) surface.
%               The IPS surface characteristics are determined by the selection of window
%               type (wintype), window length (winlen) and the distance that the window
%               is moved through the data sequence (step). The mean is then subtracted from
%               the IPS surface and is placed sequentially in the P matrix for display.
%               The P matrix plots only the positive half of the spectral plane.
%
% Input Variables:
%    data:      input data
%    wintype:  '0' Rectangular Window
%               '1' Hamming Window
%    winlen:    The desired width of the window, normally half of the siglen
%    step:      Time step desired, normally '1' or a multiple of '2'
%    p:          p = 1 indicates contour plot
%                p = 2 indicates mesh plot
%
% Output Variables:
%    P:          The IPS time-frequency surface
%
% Associated MATLAB Files: pgr.m
%
% Associated Functions: none
clg
[datarows,datacolumns] = size(data);
if datarows ~= 1
  data=data.';
end
siglen = length(data);
if wintype == 0
  win = ones(winlen-1,1);
  elseif wintype == 1
  win = hamming(winlen-1);
end
W = [win(winlen/2:-1:1)];
x = [zeros(1,winlen) data zeros(1:winlen)];
p = zeros(siglen/step,winlen);
index = 1;
for n = winlen+1:step:siglen+winlen-step+1
  Xm = [conj((x(n:-1:n-(winlen/2-1)))).' (x(n:n+(winlen/2-1))).'];
  Xn = [x(n);conj(x(n))];
  product = ((Xm*Xn).*W).';
  product = [product 0 conj(product(winlen/2:-1:2))];
  p(index,:) = fftshift(real(.5*fft(product)));
  index = index+1;
end
p = p(:,winlen/2+1:winlen);
[prow,pcolumn] = size(p);
%Smoothing
p_temp(1,:) = mean(p(1:3,:));
p_temp(2,:) = mean(p(1:4,:));
p_temp(prow-1,:) = mean(p(prow-3:prow,:));
p_temp(prow,:) = mean(p(prow-2:prow,:));
for m = 3:prow-2
  p_temp(m,:) = mean(p(m-2:m+2,:));
end
P = fliplr(p_temp);
[PGR,ii,jjjj,maxmwt] = pgr(P);
[mmm,nnn] = size(P)
freqindex = [0:pcolumn-1];
timeindex = [1:prow];
if pp == 1,
contour(P',20,timeindex*step,freqindex/2/nnn)
xlabel('time')
ylabel('fraction of sampling frequency')
end
if pp == 2,
mesh(abs(P'))
end
return
```

4. Wavelet Transforms

A. Wavelet Transform Algorithm Main Body

```
% Filename: wave.m
% Title:  Wavelet Transform Algorithm Main Body
% Date of Last Revision: 04 Dec 92
% Comments: This is the main body MATLAB routine for the
%              "Discrete" Continuous Wavelet transform (DCWT), a trous discrete
%              wavelet transform (DWT) and Mallat's DWT algorithm.
%
% Input Variables:
%    data:    Input data record must be loaded prior to running this program
%    IMAX:    Length of scale axis (IMAX=5)
%    k:       Wavelet parameter (k<pi)
%    beta:    Modulation factor for Morlet window (beta<0.5)
%    L:       Length of points in Morlet window (L=101)
%    N:       Number of points in record (1024)
%    ii:      Time bin where target appears
%    jjjj:    Scale bin where target appears
%    ip:      Plot signal & wt or wt only
%
% Output Variables:
%    wtt2:  magnitude data of scale-time plot
%
% Associated MATLAB Files: 1.) atrou_v.m
%                          2.) mallat.m
%                          3.) dcwt.m
%
%Associated Functions:    1.) pgr.m
%
pp=input('pick plot pp = 1 contour; pp = 2 mesh: ');
clg
```

```
clear xt xtt  g1 g2 g
ip = 1;
data = input('input data vector: ');
imax = input('input imax (interpolation only to i = 6): ');
a = input('pick algorithm a = 1 DWT atr; a = 2 DWT mallat; a = 3 CDWT: ');
if a == 1,
  ifunc = input('3rd/7th order preintegrator f: 3/7 :');
  if ifunc == 3, f = [0.5,1,0.5];end
  if ifunc == 7, f= [-1/16,0,9/16,1,9/16,0,-1/16]:end;
  voices = input('input number of voices:');
end
if a == 3,
  k = input('input k (k<pi): ');
  beta =i nput('input beta (beta<.5): ');
end
[np,mp] = size(data);
if np>1,data = data'; end
[np,mp]=size(data); N = mp;
m = mean(data);
xt(1,:) = (data(1,:)-m);                    %take mean of signal out
if a == 1,  [wtt2,beta,k,kk] = atrou_v(xt,imax,f,voices);end
if a == 2,  [wtt2] = mallat(xt);end
if a == 3,  [wtt2] = wavefunc(xt,k,beta,imax);end
[PGR,ii,jjjj,maxmwt] = pgr(wtt2);
[MM,NN] = size(wtt2);
if pp == 1,
if a == 1, contour(abs(wtt2).^2,10,1:N,(0:kk-2)/voices);end;
if a == 2, contour(abs(wtt2).^2,10,1:NN,0:MM-1);end;
if a == 3, contour(abs(wtt2).^2,10,1:NN,0:MM-1);end;
 title('DWT Scalogram two linear chirps (one voice a trous): PGR=',num2str(PGR),'dB,
beta=',num2str(beta)')
 xlabel('time')
 ylabel('octave')
end
if pp == 2,
 mesh(abs(wtt2).^2)
end
```

B. "Discrete" Continuous Wavelet Transform

```
function wtt=dcwt(data1,k,beta,imax)
% Filename: wavefunc.m
% Title: Wavelet Transform
% Date of Last Revision: 04 Nov 92
% Comments: This program calculates the "Discrete" Continuous
%               Wavelet Transform (DCWT).
%
% Input Variables:
%    data1:   Input data record must be loaded prior to running this program
%    k:       Wavelet parameter (k=7.5)
%    beta:    Regulating Parameter for the Morlet wavelet
%    imax:    Length of scale axis (imax <=6)
```

```
%
% Output Variables:
%      wtt:        Magnitude data of scale-time plot
%
% Associated MATLAB Files: wave.m
%
% Associated Functions: none
%
a0 = 2;
xt = data1;
[M,N] = size(xt);
imax = 6;
x = 0:N-1;
x0 = x ;
fs = 1;
m = mean(xt);
beta2 = beta^2;
fxt(1,:) = fft(xt(1,:)-m);
xg = 2*pi*x*fs/N;
for i = 0:imax-1
  a = a0^(i);
  g(1,:) = (1/beta)*exp( (-0.5/beta2)*(a*xg-k*ones(xg)).^2 );
  temp = g(1,:).*fxt(1,:);
  wtx(i+1) = i;
  wtt(imax-i,:) = sqrt(a)*ifft( temp );
end
return
```

C. *à trous* Discrete Wavelet Transform

```
function [wtt2,beta,k,kk]=atrou_v(xtt,imax,f,voices);
% Filename:  atrou_v.m
% Title:  Direct Wavelet Transform Function (a trous)
% Date of Last Revision: 04 Dec 92
% Comments: This function implements the a trous algorithm for the non-orthogonal
%                  discrete wavelet transform (DWT) with multiple voices and
%                  non-causal filters.
%
% Input Variables:
%      data:     Input data record must be loaded prior to
%                running this program.
%      IMAX:  Length of scale axis (maximum IMAX=6)
%      k:        Analysis frequency for Morlet window
%                (constraints: pi/2,2*pi*beta<=k<=pi-sqrt(2)*beta)
%      beta:     Constant proportional to the bandwidth of the
%                Morlet window  (beta=.25)
%                (constraint: beta<=k/(2*pi))
%      voices:  Number of voices in Morlet window
%      N:        Number of points in record (1024)
%      M:        Number of pulses
%
% Output Variables:
```

```
%    mwt:  An array (imax by N) containing the magnitude of
%              DWT coefficients.
%
% Associated MATLAB Files: wave.m
%
% Associated Functions: 1.)  gwin_v.m (calculates Morlet window)
%                                   2.)  interpm.m.m (inserts zeros)
j = sqrt(-1);
[M,N] = size(xtt);
if voices  >  1
   beta = pi/(4*sqrt(2)*voices);
   k = pi-sqrt(2)*beta;
else
   k = input('for voices = 1 enter k: ');
   beta = input('for voices = 1 enter beta (beta<0.5): ');
 end
L = fix(2*sqrt(2)/beta);
wtt = zeros(voices*imax,L+N);
kk = 1;
for i = 1:imax
  for vnum = 0:voices-1
     g = gwin_v(L,k,beta,i,vnum,voices);
     clear wtt1  wttemp;
     L2 = fix(L/2);
     wttemp = conv(g,xtt);
     mt = length(wttemp);
     wtt1 = wttemp(L2+1:mt);
   if i  <  2
     wtt1=wtt1;
  elseif (i > =  2  & i < =  6)
     for j = 1:i-1
       wtt1 = interpm(wtt1,2);
     end
  else
     for j = 1:6
       wtt1 = interpm(wtt1,2);
     end
  end
     wtt(kk,1:length(wtt1)) = wtt1(1:length(wtt1));
     kk = kk+1;
 end
    LF = fix(length(f)/2);
    temp = conv(f,xtt);
    nt = length(temp);
    clear xtt
    tempf = temp(LF+1:nt);
    xtt = tempf(2:2:length(tempf))/sqrt(2);
end
wtt2 = wtt(:,1:N);
wtt2 = flipud(wtt2);
return
```

D. Mallat's Discrete Wavelet Transform

```
function wtt2=mallat(data1);
% Filename:  mallat.m
% Title:  Mallat's Discrete Wavelet Transform Algorithm
% Date of Last Revision: 04 Dec 92
% Comments: This function implements Mallat's algorithm for the orthogonal discrete
%                  wavelet transform (DWT) and non-causal filters.
%
% Input Variables:
%     data1:  Input data record must be loaded prior to runnung program
%
% Output Variables:
%     wtt2:  An matrix (imax by N) containing the magnitude of  the DWT coefficients
%
% Associated MATLAB Files: wave.m
%
% Associated Functions: 1.) interpm.m
%                             2.) gmallat.m
%
 h == input('Enter number of coefficients (4 or 12): ');
 j = sqrt(-1);
 [M,N] = size(data1);
 xt = data1(1,:);
 imax = 5;
 m = mean(xt);
 xtt(1,:) = (xt(1,:)-m);
 ss = sqrt(3);
 if h == 4;  h = 1/(4*sqrt(2)).*[1+ss,3+ss,3-ss,1-ss];end;
 if h == 12; h = [.1115,.494,.7511,.3153,-.2263,-.1298,.0975,.0275,-.0316,.00056,.0048,-.0011];end;
 g = gmallat(h);
 L = length(g);
 wtt = zeros(imax,N);
 for i = 1:imax
    clear wtt1 wttemp;
    L2 = fix(L/2);
    wttemp = conv(g,xtt);
    mt = length(wttemp);
    wtt1 = wttemp(L2+2:2:mt);
    wtt1 = interpm(wtt1,2);
    size(wtt1);
   if i < 2
     wtt1 = wtt1;
   elseif (i >= 2  & i <= 6)
    for j = 1:i-1
      wtt1 = interpm(wtt1,2);
    end
   else
    for j = 1:6
      wtt1 = interpm(wtt1,2);
    end
   end
   wtt(imax-i+1,1:length(wtt1)) = wtt1(1:length(wtt1));
```

```
  temp = conv(h,xtt);
  nt = length(temp);
  L3 = fix(L/2);
  clear xtt
  xtt = temp(L3+2:2:length(temp))/sqrt(2);
end
wtt2 =wtt(1:imax,1:N);
return
```

5. Associated Functions Generic to the Main Routines

A. Processing Gain Ratio

```
function [PGR,ii,jjjj,maxmwt]=pgr(wtt)
% Filename:  pgr.m (UNIX)
% Title:  Processing Gain Ratio (PGR) Calculation
% Date of Last Revision: 04 Dec 92
% Comments: This function calculates the Processing Gain Ratio.
%
% Input Variables:
%     wwt:        Magnitude data of time-scale plot.
%
% Output Variables:
%     PGR:        Processing Gain Ratio (dB)
%     ii:         Bin where maximum value of mwt appears
%     jjjj:       Scale bin where maximum value of mwt appears
%     maxmwt:  Vector containing max points in time
%
% Associated MATLAB Files: 1.) wave.m
%                          2.) stft.m
%                          3.) wvd.m
%                          4.) ips.m
%
% Associated Functions: none
%
% normalize plot
maxwtt = abs(max(wtt));
[maxmax,ii] = max(maxwtt);
meanmean = mean(maxwtt);
mwt = abs(wtt/(maxmax));
%find max term vector in mesh plot
maxmwt = max(mwt);
%find mean of max value vector
meanmwt = mean(maxmwt);
tempmwt = maxmwt;
%replace max value with mean value
tempmwt(ii) = meanmwt;
%find mean of revised signal
meantemp = mean(tempmwt);
if (ii<100) | (ii>1000)
  PGR = 0.00;
else
```

```
  PGR = 20*log(1/meantemp)
end
%find bin where max scale occurs
xxxx = max(abs(real(fliplr(mwt'))));
jjjj = find(xxxx>.9);
return
```

B. Interpolation

```
function data=interpm(idata)
% Filename: interpm.m
% Title: Orthogonal Analyzing Wavelet
% Date of Last Revision: 04 Dec 92
% Comments: This function puts a zero between every other point
%                input array. The output array is twice as long as the input array.
%
% Input Variables:
%     idata:    The input data array  (of L points)
%
% Output Variables:
%      data:     The dilated output array (of  2XL points)
%
% Associated MATLAB Files: 1.) mallat.m
%                          2.) atrou_v.m
%
% Associated Functions: none
%
L=length(idata);l=1:2:2*L-1;
data=zeros(1,2*L-1);
data(l)=idata;
```

C. Morlet Wavelet Voices

```
function [g]=gwin_v(L,k,beta,i,vnum,voices);
%Filename: gwin_v.m
%Title: Morlet Wavelet Voices
%Date of Last Revision: 04 Dec 92
%Comments: This function calculates a multiple-voiced Morlet
%              window for discrete wavelet transform (DWT) algorithms.
%
%Input Variables:
%     L:       Number of points in window.
%     k:       Analyzing frequency for Morlet wavelet.
%     beta:    Bandwidth of Morlet window
%     voices: Number of voices.
%
%Output Variables:
%     g:       An array (of L points) containing the Morlet window
%
```

```
%Associated MATLAB Files: atrou_v.m
%
%Associated Functions: none

g = (-(L-1)/2:(L-1)/2)/((2^(1/voices))^vnum);  ;
g1(1,:) = exp( -0.5*beta^2*((x)).^2 );
g2(1,:) = exp( (j*k).*(x));
g(1,:) = g1(1,:).*g2(1,:);
return
```

D. Orthogonal Analyzing Wavelet

```
function g=gmallat(h);
% Filename:  gmallat.m
% Title:  Orthogonal Analyzing Wavelet
% Date of Last Revision: 04 Dec 92
% Comments: This function calculates the orthogonal analyzing wavelet g
%                    from the orthogonal scaling function h.
%
% Input Variables:
%     h:     Orthogonal scaling function (Daubechies coefficients)
%             L points in length.
%
% Output Variables:
%     g:     An array (of L points) containing the orthogonal
%            wavelet coefficients.
%
% Associated MATLAB Files: mallat.m
%
% Associated Functions: none
%
L=length(h);
g=zeros(1:L);
for i=0:L-1
n=i+1;
g(n)=((-1)^n)*h(L-i);
end
return
```

List of References

1. L. Cohen, "Time Frequency Distributions - a Review," *Proceedings of the IEEE*, Vol. 77, No. 7, pp. 941-981, July 1989.

2. M. J. Shensa, "The Discrete Wavelet Transform: Wedding the *à trous* and Mallat Algorithms," *IEEE Transactions on Signal Processing*, Vol. SP-40, No. 10, pp. 2464-2482, October 1992.

3. S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. II, No. 7, pp. 674-693, July 1989.

4. OSD/DARPA Ultra-Wideband Radar Review Panel, "Assessment of Ultra-Wideband (UWB) Technology," DARPA Order 6049, Battelle, Tactical Technology Center, 505 King Avenue, Columbus, Ohio, July 13 1990.

5. M. A. Pollock, "Analysis, Measurement and Experience with the Naval Ocean Systems Center Ultra-Wideband Radar Facility Report of May 1992," Doctor of Engineering Dissertation, Texas A&M University, College Station, Texas, May 1992.

6. M. I. Skolnik, *Introduction to Radar Systems*, McGraw-Hill, New York, 1980.

7. S. L. Marple, Jr. *Digital Spectral Analysis with Applications*, Prentice-Hall, 1987.

8. D. Gabor, "Theory of Communication," *Journal of the IEEE*, Vol. 93, pp. 429-457, May 1946.

9. W. Mecklenbräucker, "A Tutorial on Non-Parametric Bilinear Time-Frequency Signal Representations," *Les Houches, Session XLV, 1985*, J. L. Lacoume and R. Stora, Elsevier Science Publishers, 1987.

10. R. D. Hippenstiel and P. M. de Oliveira, "Time-Varying Spectral Estimation Using the Instantaneous Power Spectrum (IPS)," *!EEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 38, No. 10, pp. 1752-1759, October 1990.

11. S. Kadambe and G. F. Boudreaux-Bartels, "A Comparison of the Existence of Cross Terms in the Wigner Distribution and the Squared Magnitude of the Wavelet Transform and the Short Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 40, No. 10, pp. 2498-2517, October 1992.

12. R. A. Rohrbaugh, "Advanced Signal Processing Developments for Acoustic Spectral Estimation," Internal Report, David Taylor Research Center Detachment, Bremerton, Washington, March 1991.

13. O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine,* pp. 14-38, October 1990.

14. A. Grossman, R. Kronland-Martinet, and J. Morlet, "Reading and Understanding Continuous Wavelet Transforms," *Wavelets, Time-Frequency Methods and Phase Space,* Springer-Verlag, Berlin, pp. 2-20, 1989.

15. I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Communications on Pure and Applied Mathematics,* Vol. 56, no. 7, pp. 909-996, 1988.

16. P. Dutilleux, "An Implementation of the *algorithme à trous* to Compute the Wavelet Transform," *Wavelets, Time-Frequency Methods and Phase Space.* Springer-Verlag, Berlin, pp. 298-304, 1989.

17. M. Holscheidner, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian, " A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform," *Wavelets, Time-Frequency Methods and Phase Space,* Springer-Verlag, Berlin, pp. 286-297, 1989.

18. J. E. Legaspi, "One and Two Dimensional Discrete Wavelet Transforms," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1992.

19. O. Rioul and P. Duhamel, "Fast Algorithms for Discrete and Continuous Wavelet Transforms," *IEEE Transactions on Information Theory*, pp. 569-586, October 1990.

20. R. E. Parker, "Fixed and Adaptive Kernels in Cohen's Class of Time-Frequency Distributions," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1992.

21. K. Hagerman, "The Instantaneous Power Spectrum (IPS) and 1-1/2 D Instantaneous Power Spectrum Techniques ," Master's Thesis, Naval Postgraduate School, Monterey, California, June 1992.

INITIAL DISTRIBUTION LIST

	No. of Copies
Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
Library, Code 52 Naval Postgraduate School Monterey, California 93943-5000	2
Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Professor Monique P. Fargues, Code EC/Fa Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	2
Professor G. S. Gill, Code EC/Gl Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Professor Ralph D. Hippenstiel, Code EC/Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Professor Murali Tummala, Code EC/Tu Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
Professor Roberto Cristi, Code EC/Cx Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1

Naval Ocean Systems Command 1
Attn: Dr. M. Pollock, Code 755
San Diego, California 92152

Naval Ocean Systems Command 1
Attn: Dr. M. J. Shensa, Code 782
San Diego, California 92152

LT Robert E. Parker, Jr. 1
c/o Commanding Officer
Suface Warfare Officers School
San Diego, California 92155-5008

LT William A. Brooks, Jr. 1
c/o Commanding Officer
Space and Naval Warfare Command (PMW-152)
Washington, DC 20363-5000